# Installing and Configuring LiveCycle Security Products for JBoss

Adobe® LiveCycle™

September 2007

Version 7.2

© 2007 Adobe Systems Incorporated. All rights reserved.

Adobe® LiveCycle™ 7.2 Installing and Configuring LiveCycle Security Products for JBoss® for Microsoft® Windows® and Linux®
Edition 1.3, September 2007

If this guide is distributed with software that includes an end user agreement, this guide, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by any such license, no part of this guide may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Adobe Systems Incorporated. Please note that the content in this guide is protected under copyright law even if it is not distributed with software that includes an end user license agreement.

The content of this guide is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

Please remember that existing artwork or images that you may want to include in your project may be protected under copyright law. The unauthorized incorporation of such material into your new work could be a violation of the rights of the copyright owner. Please be sure to obtain any permission required from the copyright owner.

Any references to company names in sample templates are for demonstration purposes only and are not intended to refer to any actual organization.

Adobe, the Adobe logo, Acrobat, Kozuka Gothic, Kozuka Mincho, LiveCycle, Minion, Myriad, and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

BEA WebLogic Server is a registered trademark of BEA Systems, Inc.

IBM, AIX, DB2, and WebSphere are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Intel and Pentium are registered trademarks of Intel Corporation in the U.S. and other countries.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

MacIntosh is a trademark of Apple Computer, Inc., registered in the United States and other countries.

Microsoft, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Novell and SUSE are registered trademarks of Novell, Inc. in the United States and other countries.

Oracle is a trademark of Oracle Corporation and may be registered in certain jurisdictions.

Red Hat and JBoss are trademarks or registered trademarks of Red Hat, Inc. in the United States and other countries.

Sun, Java, JavaScript, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

All other trademarks are the property of their respective owners.

This product includes software developed by the Apache Software Foundation (http://www.apache.org/).

This product includes code licensed from RSA Data Security.

Portions copyright 1992, 1993 Simmule Turner and Rich Salz. All rights reserved.

This product contains either BISAFE and/or TIPEM software by RSA Data Security, Inc.

Portions of this code are licensed from Apple Computer, Inc. under the terms of the Apple Public Source License, Version 2. The source code version of these portions and the license are available at http://www.opensource.apple.com/apsl/.

This Program was written with MacApp®: ©1985-1988 Apple Computer, Inc. APPLE COMPUTER, INC. MAKES NO WARRANTIES WHATSOEVER, EITHER EXPRESS OR IMPLIED, REGARDING THE PRODUCT, INCLUDING WARRANTIES WITH RESPECT TO ITS MERCHANTABILITY OR ITS FITNESS FOR ANY PARTICULAR PURPOSE. The MacApp software is proprietary to Apple Computer, Inc. and is licensed to Adobe for distribution only for use in combination with Adobe software.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA.

Notice to U.S. Government End Users. The Software and Documentation are “Commercial Items,” as that term is defined at 48 C.F.R. §2.101, consisting of “Commercial Computer Software” and “Commercial Computer Software Documentation,” as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright laws of the United States. Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, USA. For U.S. Government End Users, Adobe agrees to comply with all applicable equal opportunity laws including, if appropriate, the provisions of Executive Order 11246, as amended, Section 402 of the Vietnam Era Veterans Readjustment Assistance Act of 1974 (38 USC 4212), and Section 503 of the Rehabilitation Act of 1973, as amended, and the regulations at 41 CFR Parts 60-1 through 60-60, 60-250, and 60-741. The affirmative action clause and regulations contained in the preceding sentence shall be incorporated by reference.

# Contents

**Preface** ..... **6**
What's in this guide? ..... 6
Who should read this guide? ..... 6
Conventions used in this guide ..... 6
Related documentation ..... 7
Updated LiveCycle product information ..... 8

**1 Before You Install** ..... **9**
About the installation, configuration, and deployment process ..... 9
Methods for installing, configuring, and deploying LiveCycle products ..... 10
Upgrading LiveCycle products ..... 11
System requirements ..... 11
Supported software ..... 11
Platform and software combinations ..... 13
Minimum hardware requirements ..... 13
Installation, configuration, and deployment checklists ..... 13
Turnkey installation and deployment checklist ..... 13
Manual installation and deployment checklist ..... 14
LiveCycle Reader Extensions Rights credential ..... 15
Preparing trust components for LiveCycle Reader Extensions ..... 16
Preparing trust components for LiveCycle Document Security ..... 17
Obtaining digital certificates and CRLs ..... 17
Configuring trust data ..... 17
Signing and validating trust.xml ..... 18
Updated LiveCycle product information ..... 18

*Part I: Turnkey Installation*

**2 Installing LiveCycle Products Using the Turnkey Installation** ..... **20**
Installing LiveCycle Reader Extensions or LiveCycle Document Security ..... 20
Modifying LiveCycle services ..... 22
Installing LiveCycle Policy Server ..... 22
Modifying LiveCycle services ..... 25
Next step ..... 25

**3 Post-deployment** ..... **26**
LiveCycle Policy Server ..... 26
Configuring LiveCycle Policy Server run-time properties ..... 26
Adding Kerberos authentication ..... 28
Network requirements ..... 28
Creating the Active Directory account ..... 28
Adding Kerberos authentication using the administration interface ..... 29
Configuring support for Novell's eDirectory directory server ..... 30
LiveCycle Reader Extensions ..... 30
LiveCycle Document Security ..... 31

*Part II: Manual Configuration and Deployment*

**4 Installing LiveCycle Products ..... 33**
Performing the installation ..... 33
Viewing the error log ..... 35
Next steps ..... 35
**5 Preparing your Environment ..... 36**
Creating the LiveCycle database ..... 36
Creating a MySQL database ..... 36
Creating an Oracle database ..... 37
Creating a DB2 database ..... 38
Creating a SQL Server database ..... 40
Preparing JBoss ..... 41
Installing JBoss Application Server 3.2.5 ..... 41
Setting up users, roles, and login files ..... 41
Configuring DocumentServicesLibrary.jar ..... 42
Copying files ..... 42
Copying files to run LiveCycle Policy Server with LiveCycle Workflow ..... 43
Installing database drivers ..... 43
Next step ..... 43
**6 Configuring LiveCycle Products ..... 44**
Creating an endorsed directory ..... 44
Configuring LiveCycle products for deployment ..... 45
Next step ..... 48
**7 Manually Configuring JBoss ..... 49**
Starting and stopping JBoss ..... 49
Setting up JBoss ..... 50
Copying the JBoss configuration files ..... 50
Configuring JBoss properties ..... 51
Configuring the transaction time-out property ..... 51
Connecting JBoss to the database ..... 52
Configuring a MySQL data source ..... 52
Configuring a SQL Server data source ..... 53
Configuring an Oracle data source ..... 53
Configuring a DB2 data source ..... 54
Configuring security ..... 55
Configuring JAAS authentication ..... 55
Sample application policy ..... 56
Example application policies ..... 58
Next step ..... 59
**8 Manually Deploying to JBoss ..... 60**
About deploying LiveCycle products ..... 60
JBoss directory name ..... 60
Summary of deployable components ..... 61
Deploying to JBoss ..... 61
Resetting the counter for LiveCycle Reader Extensions ..... 62
Viewing log information ..... 62
Next step ..... 62
**9 Initializing the Database ..... 63**
Next step ..... 63

*Part III: Post-Deployment Configuration*

**10 Configuring SSL on JBoss ..... 65**
Creating an SSL Credential ..... 66
Enabling SSL ..... 67

**11 Integrating with LiveCycle Policy Server ..... 69**
Configuring LiveCycle products to integrate with LiveCycle Policy Server ..... 69
Configuring multiple security products for integration ..... 70
Configuring LiveCycle Policy Server for EJB access ..... 70

**A Uninstalling LiveCycle Products ..... 71**
Removing the product files installed using a turnkey installation ..... 71
Removing the product files installed using manual installation ..... 72

**B Upgrading LiveCycle Products to Version 7.2 or 7.2.1 ..... 73**
Upgrade guidelines ..... 73
Updating your application server ..... 74
Using automatic or turnkey installations for upgrading ..... 74
Configuring using Configuration Manager during the upgrade process ..... 74
Summary of manual upgrade process ..... 74
LiveCycle Forms and LiveCycle Print ..... 75
Upgrading from Adobe Form Server 6.0 to LiveCycle Forms 7.2 ..... 75
LiveCycle Forms and LiveCycle Print 7.x to LiveCycle Forms and LiveCycle Print 7.2 ..... 76
LiveCycle Form Manager ..... 79
LiveCycle Assembler, LiveCycle Workflow, and Watched Folder ..... 81
LiveCycle Workflow Designer ..... 84
BAM Server for LiveCycle Workflow ..... 84
About the BAM Server upgrade utility ..... 85
LiveCycle PDF Generator ..... 86
LiveCycle Document Security ..... 89
LiveCycle Reader Extensions ..... 92
LiveCycle Policy Server ..... 95

**C Enhancing Server Performance ..... 96**
Optimizing inline documents and impacts on JVM memory ..... 96
Cleaning up temporary files from Global storage directory ..... 97

**D Supported Platform and Software Combinations ..... 98**

**E Content and Format of the trust.xml File ..... 103**
Example of the trust.xml file ..... 103
trustAnchor element ..... 104
credentials element ..... 105
CRL element (Certificate revocation) ..... 106
prefs element (Plug-in preferences) ..... 107

**F Fonts Installed with the Font Manager Module ..... 110**

**Index ..... 111**

# Preface

This guide is one of several resources available to help you learn about Adobe® LiveCycle™ suite of security products. The LiveCycle security products enable you to dynamically apply settings such as usage rights, encryption, digital signatures, and confidentiality settings to existing Adobe PDF documents.

## What's in this guide?

This guide provides information about how to install and configure the following LiveCycle security products for Microsoft® Windows®and Linux®, and how to deploy the security products to a JBoss® Application Server:

- Adobe LiveCycle Document Security 7.2
- Adobe LiveCycle Policy Server 7.2
- Adobe LiveCycle Reader Extensions 7.2

## Who should read this guide?

This guide provides information for administrators or developers responsible for installing, configuring, administering, or deploying LiveCycle security products. The information is based on the assumption that anyone reading this guide is familiar with application servers, Linux and Windows operating systems, MySQL, Oracle®, DB2®, or SQL Server databases, and web environments.

## Conventions used in this guide

This guide uses the following naming conventions for common file paths.

| Name | Default value | Description |
|---|---|---|
| *[LiveCycle root]* | Windows:<br>C:\Adobe\LiveCycle\<br>Linux:<br>/opt/adobe/livecycle/ | The installation directory that is used for all LiveCycle products. The installation directory contains subdirectories for Configuration Manager, product SDKs, and each installed LiveCycle product (along with the product documentation). |
| *[product root]* | Windows:<br>C:\Adobe\LiveCycle\components<br>Linux:<br>/opt/adobe/LiveCycle/components | The directory where product-specific directories and files (such as documentation, uninstall files, samples, and license information) are located. |

| Name | Default value | Description |
|---|---|---|
| *[appserver root]* | • Windows (Turnkey Installation): C:\Adobe\LiveCycle\jboss\<br>• Windows (Manual Installation): C:\jboss<br>• Linux: /opt/jboss | The home directory of the application server that runs the LiveCycle products. |
| *[dbserver root]* | Depends on the database type and your specification during installation. | The location where the LiveCycle database server for LiveCycle Policy Server is installed. |

Most of the information about directory locations in this guide is cross-platform (all file names and paths are case-sensitive in Linux). Any platform-specific information is indicated as required.

# Related documentation

This guide contains instructions for deploying LiveCycle security products to JBoss. The *Installing and Configuring LiveCycle Security Products* guides for other supported application servers can be accessed at: www.adobe.com/support/documentation/en/livecycle/.

The resources in this table can help you learn about and get started using LiveCycle products.

| For information about | See |
|---|---|
| General information about a product and how it integrates with other Adobe products | *Overview* guides for each product. |
| The product architecture, how to use the APIs, and how to develop custom applications for use with the product | The developer guides for each product |
| The EJB API, including descriptions and explanations of its classes and methods | The API reference for each product. Most API references are installed as JavaDocs with each product. However, some API references are provided as PDF documents with the product. |
| How to use the LiveCycle Policy Server form administration and user features | *LiveCycle Policy Server Help* |
| How to use the LiveCycle Reader Extensions web application | *LiveCycle Reader Extensions Help* |
| Other services and products that integrate with LiveCycle products | www.adobe.com |
| Patch updates, technical notes, and additional information on this product version | www.adobe.com/support/products/enterprise/index.html |

## Updated LiveCycle product information

Adobe Systems has posted a Knowledge Center article to communicate any updated LiveCycle product information with customers. You can access the article at: www.adobe.com/support/products/enterprise/knowledgecenter/c4811.pdf.

# 1 Before You Install

This chapter describes how to prepare your system for installing LiveCycle security products:


- "About the installation, configuration, and deployment process" on page 9
- "Deploying multiple LiveCycle products" on page 10
- "System requirements" on page 11
- "Installation, configuration, and deployment checklists" on page 13


Before you begin installing LiveCycle products on your application server, visit the Adobe LiveCycle product download page at the following location to make certain you have the latest version of the software:

www.adobe.com/support/products/enterprise/support_downloads.html

## About the installation, configuration, and deployment process

Installing, configuring, and deploying LiveCycle products involves the following processes:

**Installing:** Installing the products places all of the required files onto your computer, within one installation directory structure. You install the products by running the installation program. The default installation directory is C:\Adobe\LiveCycle (Windows) or /opt/adobe/livecycle (Linux); however, you can install the files to a different directory. In this guide, the installation directory is referred to as *[LiveCycle root]*. In order for multiple LiveCycle products to interoperate with one another, you must install all of the products in the same *[LiveCycle root]* location. This enables you to assemble the multiple LiveCycle products into one EAR file. (See "Installing LiveCycle Products" on page 33.)

**Configuring and assembling:** Configuring the products modifies a variety of settings that determine how the products work. Assembling the products packages all of the installed components that the products need into deployable EAR files, according to your configuration instructions. You configure and assemble the products for deployment by running Configuration Manager. (See "Configuring LiveCycle Products" on page 44.)You can configure and assemble multiple LiveCycle products at the same time.

**Deploying:** Deploying the products involves deploying the assembled EAR files and a few other configured files to the JBoss application server on which you plan to run your LiveCycle solution. If you have configured and assembled multiple products, most of the deployable components for the multiple products are packaged within the single deployable LiveCycle.ear file. (See "Manually Deploying to JBoss" on page 60.)

**Initializing the LiveCycle database**: Initializing the LiveCycle database creates the LiveCycle Policy Server tables and loads the configuration settings, the default administrator user account, and other internal user accounts. Deploying any LiveCycle product that connects to the LiveCycle database requires you to initialize the LiveCycle database after the deployment process. (See "Initializing the Database" on page 63.)

# Methods for installing, configuring, and deploying LiveCycle products

You can use one of the following methods for installing, configuring, and deploying LiveCycle products as well as initializing the database (for LiveCycle Policy Server):

**Turnkey:** The turnkey method lets you install the files, and then run Configuration Manager to configure the EAR files and other components and automatically perform the following tasks:

- Install and configure the JBoss Application Server.
- (LiveCycle Policy Server) Install and configure the MySQL database.
- Assemble and deploy LiveCycle products to JBoss.
- (LiveCycle Policy Server) Initialize the MySQL database.

The turnkey method does not configure SSL. (See "Turnkey installation and deployment checklist" on page 13 or "Installing LiveCycle Products Using the Turnkey Installation" on page 20.)

**Manual:** The manual method lets you install the files, and then run Configuration Manager to configure the EAR files and other components. You can also choose to configure your application server and deploy your EAR file manually. However, you must manually install, configure, and start your application server before running Configuration Manager and deploying to the application server. For LiveCycle Policy Server, you must also manually create and configure the database and run Configuration Manager a second time to initialize the database. (See "Manual installation and deployment checklist" on page 14.)

## Deploying multiple LiveCycle products

To deploy multiple LiveCycle products so that they interoperate, you need to install them in the same location, assemble them in multiple EAR files, and then deploy the EAR files. You can use the turnkey method to install and deploy each of the products and, using Configuration Manager, configure and assemble all of the products. Deploying multiple products is done during the configuration process with Configuration Manager.

If you are deploying LiveCycle Document Security or LiveCycle Reader Extensions with another LiveCycle product, you need to install and configure a database. For more information, see the *Installing and Configuring LiveCycle* guide.

For information on integrating LiveCycle Document Security or LiveCycle Reader Extensions with LiveCycle Policy Server, see "Integrating with LiveCycle Policy Server" on page 69.

If you have already installed and deployed other LiveCycle products, you must use the same name for the new EAR files as the previous EAR files. If you use a different name, you need to undeploy the other products before deploying the new EAR files.

**Note:** To install multiple products in the LiveCycle 7.2 product suite, install in the following order:

- LiveCycle PDF Generator
- LiveCycle Assembler, LiveCycle Forms, LiveCycle Form Manager, LiveCycle Print, LiveCycle Workflow
- Watched Folder
- LiveCycle Document Security, LiveCycle Policy Server, LiveCycle Reader Extensions

For detailed documentation on installing multiple products, refer to the Knowledge Center article at: www.adobe.com/support/products/enterprise/knowledgecenter/c4811.pdf.

# Upgrading LiveCycle products

For information on upgrading LiveCycle products, refer to "Upgrading LiveCycle Products to Version 7.2 or 7.2.1" on page 73 in this guide.

# System requirements

This section includes details about the software and hardware that is required for running LiveCycle products.

**Note:** If you are running the turnkey installation and configuration, the only prerequisite software requirement is the Java™ Development Kit (JDK) and, optionally, a web browser. The turnkey method installs and configures JBoss (which includes the Apache web server) and a MySQL database (for LiveCycle Policy Server).

## Supported software

This table provides a summary of the application servers, web servers, web browsers, and JDK versions that LiveCycle products support. For a complete list, see "Supported Platform and Software Combinations" on page 98.

| Required software | Supported version |
|---|---|
| Operating System | • Microsoft Windows Server™ 2003 Enterprise Edition or Standard Edition with Service Pack 1<br>• SUSE™ Linux Enterprise Server 9.0 i386 (32-bit)<br>• (LiveCycle Policy Server) Red Hat® Linux Advanced Server 2.1 Update 3 or 3.0 |
| Application server | • JBoss Application Server 3.2.5 |
| Web browser | • Microsoft Internet Explorer 6.0 for Windows<br>**Note:** LiveCycle Reader Extensions supports only Microsoft Internet Explorer 6.0<br>• Netscape 7.1 or higher for Windows<br>• Netscape 7.2 or higher for Linux<br>• Mozilla 1.8 or higher for Windows and Linux<br>• Safari 1.2.3, Safari 1.3, Safari 2.0 (end-user support for Macintosh only) |
| JDK | • J2SDK version 1.4.2_04 or later (version 1.4.2_10 is not supported)<br>You must create or set the `JAVA_HOME` environment variable to point to the location where Java is installed. Ensure that the following is set:<br>`Set PATH=%JAVA_HOME%/bin;%PATH%` (Windows) or `Set PATH=$JAVA_HOME/bin:$PATH` (Linux). |

| Required software | Supported version |
|---|---|
| Database | • MySQL 4.1<br>If you want to set up a database schema and new users on MySQL using a graphical user interface (GUI), you must install the MySQL Administrator tool. For more information, see the MySQL user documentation.<br>• IBM DB2 8.2 (Version 8.1 FixPack 7)<br>• Oracle 9i<br>• Oracle 10g<br>• MS SQL Server 2000 SP 3<br>**Note:** A database is only required for LiveCycle Policy Server. |
| Database driver | • MySQL - mysql-connector-java-3.0.15-ga-bin.jar<br>• IBM DB2 - db2cc.jar, db2cc_license_cu.jar<br>• Oracle 10g - ojdbc14.jar version 10.1.0.4<br>• MS SQL Server 2000 - msbase.jar, mssqlserver.jar, msutil.jar<br>• MySQL 4.1 - mysql-connector-java.jar<br>**Note:** Database drivers are only required for LiveCycle Policy Server. |
| LDAP server | • Sun ONE 5.1, 5.2<br>• Microsoft Active Directory 2000<br>• Microsoft Active Directory 2003<br>• Novell® eDirectory 8.7<br>**Note:** An LDAP server is only required for LiveCycle Policy Server and LiveCycle Reader Extensions. |
| PDF client | • Adobe Acrobat® 7.0 Professional and Acrobat 7.0 Standard (for securing documents with policies and opening policy-protected documents)<br>• Adobe Reader® 7.0 (for opening policy-protected documents) |

**Note:** The following software is provided on the LiveCycle Policy Server DVD for your convenience:

- Oracle 9i thin client (type 2)
- MySQL Connector/J version 3.0
- DB2 JDBC driver
- JBoss-3.2.5_tomcat_4.1.24

**Note:** If you run the turnkey installation, you must not have instances of JBoss Application Server or the MySQL database server running on the target computer.

## Platform and software combinations

The table in this section summarizes the software combinations supported for each operating system. For a complete list, see "Supported Platform and Software Combinations" on page 98. The database information applies only to LiveCycle Policy Server. This table provides a summary of the operating system and database combinations supported with JBoss 3.2.5.

| Operating system | Database |
|---|---|
| Red Hat Linux Advanced Server 3.0 | MS SQL Server 2000 SP 3 |
| Windows Server 2003 Enterprise Edition or Standard Edition | MySQL 4.1 |

**Note:** LiveCycle Policy Server is multilingual (supporting English, French, German and Japanese) when used with the Oracle, DB2, and SQL Server databases.

## Minimum hardware requirements

The table in this section lists the supported operating systems and corresponding hardware. For any installation, the following settings are recommended as a minimum:

- Disk space for installation: 3 GB per product
- System temp space during installation: 2 GB
- Memory for running the products: 1 GB per product for each CPU

| Operating system | Minimum hardware requirement |
|---|---|
| Windows Server 2003 Enterprise Edition or Standard Edition | Intel® Pentium® 3 or x86 equivalent, 1GHz processor |
| SUSE Linux Enterprise Server 9.0 i386 (32-bit) | Pentium 3 or x86 equivalent, 1GHz processor |
| Red Hat Linux Advanced Server 2.1 or 3.0 | Pentium 3 or x86 equivalent, 1GHz processor |

# Installation, configuration, and deployment checklists

This section includes checklists that you can use to step through the installation and configuration process. A checklist is provided for installing and configuring when using either the turnkey method or the manual method.

Before starting the installation, ensure that the JAR files are not associated with WinZip or any other application other than the java application launcher.

## Turnkey installation and deployment checklist

The following table includes the steps required for installing LiveCycle products using the turnkey method. The turnkey installation automatically performs all of the tasks required to install and configure LiveCycle products on a JBoss Application Server running on Windows.

Perform this type of installation and configuration if you do not yet have an application server installed and configured and you want to use JBoss, or if you want to quickly and easily get the products installed and configured for testing or demonstration purposes.

Do not use the turnkey configuration method if you plan to enable SSL. If you require SSL, perform a manual configuration. (See "Installing LiveCycle Products" on page 33.)

| | Task | Topic |
|---|---|---|
| ☐ | Ensure that you have the required software installed in the target environment. | "System requirements" on page 11 |
| ☐ | Run the installation program with the turnkey option enabled for JBoss.<br>For LiveCycle Policy Server, the product, JBoss, and MySQL are installed.<br>For LiveCycle Reader Extensions and LiveCycle Document Security, the product and JBoss are installed. | "Installing LiveCycle Products Using the Turnkey Installation" on page 20 |
| ☐ | (LiveCycle Reader Extensions) Set up default users, roles, and login files. | "Setting up users, roles, and login files" on page 41 |
| ☐ | (LiveCycle Policy Server) Configure JAAS authentication. | "Configuring JAAS authentication" on page 55 |
| ☐ | Access the LiveCycle Reader Extensions or LiveCycle Policy Server web application. | "Post-deployment" on page 26 |
| ☐ | Check the log file. | "Viewing log information" on page 62 |

## Manual installation and deployment checklist

The following table includes the steps required for installing LiveCycle products using the manual method. Your application server must be installed before you perform the installation.

Perform this type of installation if you are installing the product in a production environment.

**Note:** If you are installing multiple products, ensure that they are all installed before running Configuration Manager to configure and deploy them.

| | Task | Topic |
|---|---|---|
| ☐ | Ensure that you have the required software installed in the target environment. | "System requirements" on page 11 |
| ☐ | Run the installation program. | "Installing LiveCycle Products" on page 33 |
| ☐ | (LiveCycle Policy Server) Create the database and configure the application server. | "Preparing your Environment" on page 36 |
| ☐ | (LiveCycle Reader Extensions, LiveCycle Document Security) Complete the application server configuration. | "Preparing your Environment" on page 36 |

| | Task | Topic |
|---|---|---|
| ☐ | Run Configuration Manager and select the Custom Configuration Wizard. This will configure and assemble the products. | "Configuring LiveCycle Products" on page 44 |
| ☐ | Configure the JBoss settings. A variety of settings must be configured. | "Manually Configuring JBoss" on page 49 |
| ☐ | (LiveCycle Policy Server) Configure JAAS authentication. | "Configuring JAAS authentication" on page 55 |
| ☐ | Deploy the product deployment files to the application server. | "Manually Deploying to JBoss" on page 60 |
| ☐ | (LiveCycle Policy Server) Run Configuration Manager to initialize the LiveCycle database. | "Initializing the Database" on page 63 |
| ☐ | Access the LiveCycle Reader Extensions or LiveCycle Policy Server web application. | "Post-deployment" on page 26 |
| ☐ | Check the log file. | "Viewing log information" on page 62 |
| ☐ | Configure SSL on the application server, if required. | "Configuring SSL on JBoss" on page 65 |

# LiveCycle Reader Extensions Rights credential

Before you begin the installation process, you must obtain and install the LiveCycle Reader Extensions Rights credential. This credential is a digital certificate specific to LiveCycle Reader Extensions that enables Adobe Reader usage rights to be activated in the PDF documents produced. If the credential is not installed, LiveCycle Reader Extensions will not install properly or run. You cannot use a standard digital certificate for this function; you must use the dedicated Rights credential.

**Note:** The Rights credential cannot be used for typical document signing or assertion of identity. For these applications, you can use a self-sign certificate or acquire an identity certificate from a Certificate Authority (CA).

## Obtaining a Rights credential

The following types of Rights credentials are available:

**Customer Evaluation**: Creates draft documents only and is valid for 2-3 months.

**Production**: Full-functioning certificate provided to customers who have purchased the full product. Production certificates are unique to each customer but can be installed on multiple systems.

The Rights credential is delivered as a digital certificate containing both the public key and the private key, and the password used to access the credential.

If your organization orders an evaluation version of LiveCycle Reader Extensions, you receive an evaluation Rights credential, either from the sales representative through whom you ordered the product or from the website where you downloaded the evaluation product. If you are already running Adobe Reader

Extensions Server 6.1 or higher, your credential is still valid and, therefore, you do not receive a new one from Adobe.

If your organization purchases a production version of LiveCycle Reader Extensions, the production Rights credential is delivered by Electronic Software Download (ESD), and you are notified by email. A production Rights credential is unique to your organization and can enable the specific usage rights that you require.

If you obtained LiveCycle Reader Extensions through a partner or software provider who has integrated LiveCycle Reader Extensions into their software, the Rights credential is provided to you by that partner who, in turn, receives this credential from Adobe.

The Rights credential is used to extend the usage rights of each PDF file processed by LiveCycle Reader Extensions. It is the most important part of the software licensing and should be stored carefully in a secure environment.

# Preparing trust components for LiveCycle Reader Extensions

If you have not yet set up a trust directory to contain your LiveCycle Reader Extensions Rights credentials, the installation program leads you through the process of setting up a trust directory and populates it with the credential you will be using to apply usage rights to PDF documents. The installation program creates a corresponding trust.xml file and places these components in the *[product root]*/trust directory.

It also configures the credential that is deployed to the application server so that the deployed product can access the credential information.

If your credential information changes, you must re-sign the trust.xml file, as well as reconfigure and redeploy the credential. You can update the trust.xml file, re-sign the trust.xml file, and redeploy the credential automatically using Configuration Manager. (See "Configuring LiveCycle Products" on page 44.)

During LiveCycle Reader Extensions configuration, if you use an existing trust directory, the following components are required.

| Trust component | Description |
|---|---|
| trust.xml | The trust.xml file contains mapping information for the certificates, credentials, and CRLs used by the PDF Manipulation Module. This file references the contents of the credentials, certificates, and CRLs directories. |
| LiveCycle Reader Extensions Rights credential | A credential obtained from Adobe that enables you to apply Adobe Reader usage rights to PDF documents. This credential is required for LiveCycle Reader Extensions to run. |
| keystore file | The keystore file stores private keys and their associated public key certificates. You create the keystore, which is used for validating the trust.xml file against the trust.sig file. It can be located anywhere on your system, but its properties are configured and maintained within the Trust Manager Module. |
| key pair | The private and public key generated and stored in the keystore is used for signing and validating the trust.xml file. This key pair is separate from the credentials and certificates described above. It is used to protect the integrity of the trust data and is used only during the product startup to verify the data integrity. |

# Preparing trust components for LiveCycle Document Security

LiveCycle Document Security uses a variety of security resources to sign, certify, encrypt, decrypt, and validate PDF documents. You can perform the most basic security operations, such as password encryption, without using certificates. However, you will need one key pair configured for basic signing operations, such as signing and validating the trust.xml file.

If you are using a public key infrastructure (PKI), you use the credentials (private key), certificates (public keys), and CRLs that make up the PKI to perform security operations with LiveCycle Document Security. You should obtain these resources before you perform the product installation so that you can configure the trust components during the installation process.

## Obtaining digital certificates and CRLs

Digital certificates are obtained from a Certificate Authority (CA) and sent to you by email or over the web as a certificate file. This certificate file contains the public keys (also called *certificates*) and references to private keys (also called *credentials*) used for encrypting and signing documents. Certificates do not contain actual private keys; instead, they contain a reference to the identity of the user who keeps the private key securely stored in an encrypted file or Hardware Security Module (HSM).

You can use Internet Explorer (Windows) to export PFX, P12, and CER files for certificates stored in any compatible certificate store available on your computer. PFX files can only be exported as allowed by the certificate store or the credential itself. CER files holding the public key corresponding to a credential can also be exported from PFX files using either Internet Explorer or OpenSSL.

The CRL distribution point describes where you can download the CRL that corresponds to a particular CER or PFX file.

The following file types are supported:

**Certificates**: DER-encoded X.509 and base64 -encoded certificate (.cer) files. Certificates verifying the trust.xml file can be either DER-encoded or base64-encoded.

**Credentials**: PKCS#12 files (.pfx files), PKCS #11 files, MSCAPI records.

**CRLs**: RFC3280.crl files.

Maintaining the security of private keys (credentials) is critical to ensuring the stability of sensitive information. A physical storage device (often called a *Hardware Security Module*) typically provides the maximum level of security for private keys. If you do not use a physical device, it is important to store highly sensitive private keys and certificates in encrypted files in a safe place.

LiveCycle Document Security supports the industry-standard PKCS #11 interface to communicate with HSMs. An HSM vendor can provide the resources and tools you need to install and configure an HSM storage system.

## Configuring trust data

If you have not yet set up a trust directory to contain your credentials, certificates, and CRLs, the installation program leads you through the process of setting up a trust directory and populates it with the credential, certificate, and CRL files you will be using to encrypt or apply digital signatures to PDF documents. The installation program creates a corresponding trust.xml file and places all of these components in the root installation directory. It also signs the trust.xml file (after allowing you to verify it) and loads it into the Trust Manager Module, which you deploy to the application server as part of the deployment process.

If you are upgrading from Document Security Server 6.0 or later, you can use your existing trust directory and trust.xml file; you can specify the existing trust directory with Configuration Manager.

This table describes the trust or security components required to run LiveCycle Document Security.

| Trust component | Description |
|---|---|
| trust.xml | The trust.xml file contains mapping information for the certificates, credentials, and CRLs used by the PDF Manipulation Module. This file references the contents of the credentials, certificates, and CRLs directories. |
| credentials | Credentials are the private keys used to establish identity in encryption operations. Credential files used with the Trust Manager Module must be stored in the credentials directory and referenced in the trust.xml file. |
| certificates | Certificates are the public keys that correspond to credentials. Certificates used with the Trust Manager Module must be stored in the certificates directory and referenced in the trust.xml file. Certificates are called *trustAnchors* in the trust.xml file. |
| CRLs | CRLs contain a list of all of the certificates that are no longer valid. The CRLs directory can be located anywhere on your system, but it is convenient to maintain it in the same location as your other trust security resources. CRLs used with the Trust Manager Module must be stored in the CRLs directory and referenced in the trust.xml file. CRL files must also be imported into the Trust Manager Module. |
| keystore file | The keystore file stores private keys and their associated public key certificates. You create the keystore, which is used for validating the trust.xml file against the trust.sig file. It can be located anywhere on your system, but its properties are configured and maintained within the Trust Manager Module. |
| key pair | The private and public key generated and stored in the keystore is used for signing and validating the trust.xml file. This key pair is separate from the credentials and certificates described above. It is used to protect the integrity of the trust data and is used only during the product startup to verify the data integrity. |

## Signing and validating trust.xml

After the installation program creates the trust.xml file and populates it with all of the trust information that references certificates (trustAnchors), credentials, and CRLs, it signs it to ensure that it is valid and protected. Any time you add credentials to your system, you must update the trust.xml file and re-sign it. The private key is used for signing, and the public key is for validation (or verification). Each time you modify the content of the trust.xml file, you must re-sign the file.

You update the trust.xml file automatically using Configuration Manager. (See "Configuring LiveCycle Products" on page 44.)

## Updated LiveCycle product information

Adobe Systems has posted a Knowledge Center article to communicate any updated LiveCycle product information with customers. You can access the article at:
www.adobe.com/support/products/enterprise/knowledgecenter/c4811.pdf.

# Part I: Turnkey Installation

This section of the guide describes how to complete a turnkey installation of LiveCycle products

For information on the manual configuration and deployment of the products, see "Manual Configuration and Deployment" on page 32.

# 2 Installing LiveCycle Products Using the Turnkey Installation

This chapter describes how to install LiveCycle products using the turnkey method. The turnkey installation automatically performs all of the tasks required to install and configure LiveCycle products on a JBoss Application Server running on Windows.

The turnkey installation performs the following tasks:

- Installs the product files
- Installs a preconfigured version of JBoss 3.2.5 (with Apache web server embedded)
- Starts Configuration Manager
- Configures and assembles the LiveCycle product components
- Deploys all of the required components to JBoss

For LiveCycle Policy Server, the turnkey installation also installs and initializes the MySQL 4.1 database.

## Installing LiveCycle Reader Extensions or LiveCycle Document Security

Before running the turnkey installation, ensure that the environment where you are installing and deploying LiveCycle products meets the system requirements. (See the note under "System requirements" on page 11). This includes manually installing the required version of the JDK. During the turnkey installation, you will be prompted for the JDK root directory. The turnkey installation then installs JBoss and the Apache web server automatically.

The turnkey installation specifies "localhost" as the host and "8080" as the port for use by JBoss. If JBoss is already installed, ensure that it is not using port 8080. You cannot configure an alternative host or port for JBoss during the turnkey installation and configuration process.

By default, the turnkey installation places the LiveCycle product and all of the related components and software in the \Adobe\LiveCycle\ directory (referred to as the *[LiveCycle root]* directory).

JBoss is installed and run from the *[LiveCycle root]*/jboss directory.

The turnkey installation creates the "JBoss for Adobe LiveCycle" Windows service.

This service is used by Configuration Manager during the turnkey installation. The service can be stopped and started using the Services window in the Administrative Tools area of the Windows Control Panel.

**Note:** If JBoss is already installed, you must stop the service before running the turnkey installation.

For the turnkey installation, it is recommended that you accept the default configuration options. If you prefer to set all of the configuration options, run the manual installation, configuration, and deployment process. For information about the configuration options, see "Configuring LiveCycle Products" on page 44.

**Tip:** To improve the speed of installation, disable any on-access virus scanning software for the duration of the installation.

➤ **(LiveCycle Reader Extensions, LiveCycle Document Security) To run the turnkey installation:**

1. At the root level of the installation DVD, double-click the .exe file.
2. On the Welcome screen, click **Next**.
3. Review the information on the Upgrading screen, and then click **Next**.
4. Type the serial number in the text box and click **Next**.
5. Read the Product License Agreement, select **I accept the terms of the license agreement**, and then click **Next**.
6. Select **JBoss**, select **Configure and deploy the product automatically**, and then click **Next**.
7. Read the license information associated with installing JBoss, select **I accept the terms of the license agreement**, and then click **Next**.
8. Read the license information associated with installing MySQL, select **I accept the terms of the license agreement**, and then click **Next**.
9. Type the path to your JDK or click **Browse** to navigate to its root directory, and then click **Next**.
10. Accept the default Adobe directory location as listed or click **Browse** and navigate to the directory where you want to install the product, and then click **Next**.

**Tip:** If you type in the name of a directory that does not exist, InstallShield creates the directory for you.

11. Review the installation details and click **Install**. A summary screen appears when the installation program finishes installing the product.
12. On the summary screen, you have the following options:
    - If you are installing only one LiveCycle product or this is the last LiveCycle product you are installing, select **Launch the Configuration Manager** and click **Finish**. Proceed to step 13.
    - If you plan to install additional LiveCycle products, deselect **Launch the Configuration Manager** and click **Finish** to exit the installation program and run the installation programs for the additional products.
13. On the Welcome screen, click **Next**.
14. Select **Typical Configuration Wizard** and click **Next**.
15. If prompted, on the Configuration Preferences screen, select either **Use last entered values** or **Revert to the default values**, and then click **Next**.

    **Note:** This screen only appears if you have previously run Configuration Manager.
16. If required, select **JBoss** as the type of application server that you are using.
17. Select **Foundation** and the products that you want to configure and deploy, and then click **Next**.
18. Review the configuration summary information, and then click **Back** to change any settings or click **Next** to continue. When you continue, Configuration Manager configures the LiveCycle product properties using the default values and assembles the products into the LiveCycle.ear file.
19. On the Trust Directory Selection screen, select **Create a new trust directory** and click **Next**.

20. Type the path to your JDK or click **Browse** to navigate to its root directory, and then click **Next**.

21. (LiveCycle Document Security) On the New Trust Directory screen, click **Browse** and navigate to the directories containing your CRLs, credentials, and certificates, and then click **Next**. If you do not want to include copies of your CRLs, credentials, and certificates in the trust directory, click **Next**.

22. On the Key Store Selection screen, select **Create a new keystore and keypair** and click **Next**.

23. On the Keystore Creation screen, type a password, alias, and key password, and then click **Next**.

24. On the Keystore Creation - DN Values screen, complete the DN registration form and click **Next**.

25. (LiveCycle Reader Extensions) On the Reader Extensions Credential Selection screen, click **Browse** to navigate to your credential file, in the **Credential Password** box, type your password, and then click **Next**.

    If you do not have a Rights credential, see "LiveCycle Reader Extensions Rights credential" on page 15, or contact your Adobe Sales Representative or Adobe Customer Support.

26. Review the configuration summary information and then click **Back** to change any settings or click **Next** to continue. When you continue, Configuration Manager configures the LiveCycle product properties using the default values and assembles the products into the LiveCycle.ear file.

27. On the Deployment Instructions screen, click **Next**.

28. On the Tasks Completed screen, click **Finish**.

    The product is now deployed to JBoss, and JBoss should be running. If JBoss is not running, you can start the JBoss for Adobe LiveCycle service by using the Windows Service utility.

    The JBoss log file (server.log) is located in the *[LiveCycle root]*/jboss/server/all/log directory. This file contains information on JBoss activity and progress.

## Modifying LiveCycle services

By default, the turnkey installation configures the JBoss for Adobe LiveCycle service to require manual startup. If you want to set the service to start up automatically, perform the following task.

➤ **To modify LiveCycle services:**

1. Click **Start** > **All Programs** > **Administrative Tools** > **Services**.

2. Right-click the "JBoss for Adobe LiveCycle" service and click **Properties**.

3. From the **Startup Type** menu, select **Automatic**, and then click **OK**.

# Installing LiveCycle Policy Server

Before running the turnkey installation, ensure that the environment where you are installing and deploying meets the system requirements. (See the note under "System requirements" on page 11). This includes manually installing the required version of the JDK. During the turnkey installation, you will be prompted for the JDK root directory. The turnkey installation will then install JBoss, MySQL, and the Apache web server automatically.

The turnkey installation specifies "localhost" as the host and port "8080" for use by JBoss and port "3306" for MySQL. If JBoss and MySQL are already installed, ensure that they are not using these ports. You cannot configure an alternative host or port for JBoss or MySQL during the turnkey installation and configuration process.

By default, the turnkey installation places the LiveCycle product and all of the related components and software in the \Adobe\LiveCycle\ directory (referred to as the *[LiveCycle root]* directory).

JBoss and MySQL are installed and run from the *[LiveCycle root]*/jboss and *[LiveCycle root]*/mysql directories.

The turnkey installation creates the following Windows services:

- "JBoss for Adobe LiveCycle"
- "MySQL for Adobe LiveCycle"

These services are used by Configuration Manager during the turnkey installation. The services can be stopped and started using the Services window in the Administrative Tools area of the Windows Control Panel.

**Note:** If JBoss and MySQL are already installed, you must stop these services before running the turnkey installation.

For the turnkey installation, it is recommended that you accept the default configuration options. If you run the manual installation, configuration, and deployment process, you can set all of the configuration options. For information about the configuration options, see "Configuring LiveCycle Products" on page 44.

**Tip:** To improve the speed of installation, disable any on-access virus scanning software for the duration of the installation.

**Note:** After running the LiveCycle Policy Server turnkey installation and configuration, you need to manually configure JAAS authentication. (See "Configuring JAAS authentication" on page 55.)

➤ **(LiveCycle Policy Server) To run the turnkey installation:**

1. At the root level of the installation DVD, double-click the .exe file.
2. On the Welcome screen, click **Next**.
3. Type the serial number in the text box and click **Next**.
4. Read the Product License Agreement, select **I accept the terms of the license agreement**, and then click **Next**.
5. Select JBoss, select **Configure and deploy the product automatically**, and then click **Next**.
6. Read the license information associated with installing JBoss, select **I accept the terms of the license agreement**, and then click **Next**.
7. Read the license information associated with installing MySQL, select **I accept the terms of the license agreement**, and then click **Next**.
8. Type the path to your JDK or click **Browse** to navigate to its root directory, and then click **Next**.

9. Accept the default Adobe directory location as listed or click **Browse** and navigate to the directory where you want to install the product, and then click **Next**.

**Tip:** If you type in the name of a directory that does not exist, InstallShield creates the directory for you.

10. Review the installation details and click **Install**. A summary screen appears when the installation program finishes installing the product.

11. On the summary screen you have the following options:
    - If you are installing only one LiveCycle product or this is the last LiveCycle product you are installing, select **Launch the Configuration Manager** and click **Finish**. Proceed to step 12.
    - If you plan to install additional LiveCycle products, deselect **Launch the Configuration Manager** and click **Finish** to exit the installation program and run the installation programs for the additional products.

12. On the Welcome screen, click **Next**.

13. Select **Typical Configuration Wizard** and click **Next**.

14. If prompted, on the Configuration Preferences screen, select either **Use last entered values** or **Revert to the default values**, and then click **Next**.

    **Note:** This screen only appears if you have previously run Configuration Manager.

15. Select **LiveCycle Policy Server**, and then click **Next**.

16. Review the configuration summary information, and then click **Back** to change any settings or click **Next** to continue. When you continue, Configuration Manager configures the LiveCycle product properties using the default values and assembles the products into the LiveCycle.ear file.

17. On the Application Configuration and Assembly Summary screen, click **Next**.

18. On the Configure and Assemble Products Summary screen, click **Next**.

19. Deploy the product by clicking **Next**.

20. In the **Application Server Base URL** box, type the URL that Configuration Manager can use to connect to the application server. The URL must be in the following format:

    `http://[host name]:[port number]`

    where `[host name]` is the name or IP address of the computer that hosts the application server, and `[port number]` is the HTTP service port that the application server uses. For example, `http://localhost:8080`.

21. On the Prepare Database screen, click **Next**.

22. On the Database Initialization screen, click **Initialize Database Now** to connect your LiveCycle products to the database, and then click **Next**.

23. On the Tasks Completed screen, click **Finish**.

    The product is now deployed to JBoss, and JBoss should be running. If JBoss is not running, you can start the JBoss for Adobe LiveCycle service by using the Windows Service utility.

    The JBoss log file (server.log) is located in the *[LiveCycle root]*/jboss/server/all/log directory. This file contains information on JBoss activity and progress.

## Modifying LiveCycle services

By default, the turnkey installation configures the JBoss for Adobe LiveCycle service to require manual startup. If you want to set the service to start up automatically, perform the following task.

➤ **To modify LiveCycle services:**

1. Click **Start** > **All Programs** > **Administrative Tools** > **Services**.
2. Right-click the "JBoss for Adobe LiveCycle" service and click **Properties**.
3. From the Startup Type menu, select **Automatic**, and then click **OK**.
4. Right-click the "MySQL for Adobe LiveCycle" service and click **Properties**.
5. From the **Startup Type** menu, select **Automatic**, and then click **OK**.

# Next step

For LiveCycle Policy Server you must configure JAAS configuration. (See "Configuring JAAS authentication" on page 55.)

For LiveCycle Document Security and LiveCycle Reader Extensions, you can now run samples, finalize the product configuration, and develop applications for your LiveCycle products. (See "Post-deployment" on page 26.)

# 3 Post-deployment

This chapter describes how to get started using your LiveCycle security products:


- "LiveCycle Policy Server" on page 26
- "LiveCycle Reader Extensions" on page 30
- "LiveCycle Document Security" on page 31


## LiveCycle Policy Server

After deploying LiveCycle Policy Server to the application server, you can perform the following tasks:

- Configure the LiveCycle Policy Server run-time properties using the LiveCycle Policy Server web pages, after logging in as an administrator. (See "Configuring LiveCycle Policy Server run-time properties" on page 26.)
- Configure the network for LiveCycle Policy Server to support the Kerberos authentication system and specify the Kerberos run-time properties. (See "Adding Kerberos authentication" on page 28.)
- Configure LiveCycle Policy Server to support Novell's eDirectory directory server. (See "Configuring support for Novell's eDirectory directory server" on page 30.)

For detailed information on how to complete the configuration tasks, including specifying the Kerberos run-time properties, see *LiveCycle Policy Server Help*.

This section assumes that you have installed the products, configured the application server, deployed the products, and initialized the LiveCycle database.

### Configuring LiveCycle Policy Server run-time properties

After the application server is configured and LiveCycle Policy Server is running, you need to configure run-time properties using the LiveCycle Policy Server web pages.

Two web application interfaces are provided that enable you to access LiveCycle Policy Server through a web browser. One interface is for administrators and one is for users. To initially log in and configure LiveCycle Policy Server, you must use the Super Administrator account that is created when you initialized the database.

The Super Administrator account has the following properties:

**User name:** administrator

**Password:** password

**Note:** The first configuration step you must perform is to change the password for the Super Administrator account.

➤ **To configure run-time properties:**

1. In a web browser, navigate to http://*hostname:port*/edc/Main.do and log in as the Super Administrator.

   Where *port* is the port number that the application server uses (the default value is 8080), and *hostname* is the name of the computer that hosts the application server.

2. Complete the tasks in the table below. For detailed information on completing each task, see *LiveCycle Policy Server Help*.

| Configuration task | Description |
|---|---|
| Change the default administrator account password | Use the Administration configuration page in the web application to change the default administrator login password. |
| Configure client and server configuration settings | Set up client and server configuration properties, including a base LiveCycle Policy Server URL and anonymous user access properties. |
| Configure default settings for offline document security | Set the offline leasing period and specify a key rollover schedule. |
| Configure authentication providers | Create a domain for your directory servers to set up user authentication.<br>For JBoss deployments, the JAAS application name must be the name of the application policy you defined in the login-config.xml file. The directory properties you specify must match those in the login-config.xml file. (See "Configuring JAAS authentication" on page 55.)<br>To add Kerberos authentication, you must have Microsoft Active Directory 2000 or 2003 controlling a domain where both the end user and the end user's computer are in the domain. (See "Adding Kerberos authentication" on page 28.) |
| Synchronize directories | Perform a directory synchronization to pull user accounts from the directory servers to the LiveCycle Policy Server database for authentication purposes. |
| Configure external user registration | Enable external user registration. This includes external user registration email, outgoing (SMTP) server information, users and groups to include or exclude, and external user access properties. |
| Add administration accounts | (Optional) Add more administration accounts for additional users who will be administering LiveCycle Policy Server on a continuing basis. |
| Back up the LiveCycle Policy Server configuration | (Optional) Export the LiveCycle Policy Server configuration settings to a file to back up your configuration. If required, you can import the settings back into LiveCycle Policy Server to restore the configuration. |

# Adding Kerberos authentication

To use Kerberos authentication with LiveCycle Policy Server, you must complete the following tasks:

- Ensure that your network meets certain network configuration criteria. (See "Network requirements" on page 28.)
- Create a special account for Kerberos in Microsoft Active Directory. (See "Creating the Active Directory account" on page 28.)
- From the administration interface, Configure LiveCycle Policy Server to work with Kerberos. For information on how to configure LiveCycle Policy Server, see *LiveCycle Policy Server Help* and for additional information see, "Adding Kerberos authentication using the administration interface" on page 29.

## Network requirements

When using Kerberos as an authentication mechanism with LiveCycle Policy Server, Acrobat 7.0 users running Windows can access LiveCycle Policy Server without having to type a user name or password. Kerberos can leverage the Windows infrastructure to determine the identity of a user who has logged into a computer without requiring that user to enter a user name and password again.

To use the Kerberos authentication system with LiveCycle Policy Server, you must be using Microsoft Active Directory 2000 or 2003, to control a domain which end users and their computers access. Then, when LiveCycle Policy Server client software in Acrobat 7.0 uses Kerberos to perform authentication, it specifies which service should receive the end user's credentials. This ensures that only LiveCycle Policy Server—referred to as *the service* in Kerberos— receives the credentials, and not some other server, service, or user. LiveCycle Policy Server can accomplish this through a special Active Directory account that is created to act as the service.

## Creating the Active Directory account

The following example shows how to create an account that Kerberos can use as a service to give LiveCycle Policy Server users access to their accounts without having to enter their user names and passwords. The steps provided in the example can be completed after LiveCycle Policy Server is already running.

### Example 3.1 *To create an Active Directory account*

If you have an existing Active Directory instance running on a server named test.2003.policyserver.net, its IP address is resolvable by a DNS server running on 192.168.1.1 and test is running the Active Directory domain entitled 2003.policyserver.net.

➤ **To create an Active Directory account on the test.2003.policyserver.net server:**

1. On test, run the Active Directory Users & Groups program (**Administrative Tools** > **Active Directory Users and Computers**). Create a new account and call it *PolServerKerberos*. To create a new account, right-click the folder called **Users** in the hierarchy and then select **New** > **User**. You are first prompted for user first/last/login name. The login name is required; the remaining properties are optional for LiveCycle Policy Server. However, Active Directory treats all of the fields as required fields.

   **Note:** The PolServerKerberos naming matches the example provided in the *LiveCycle Policy Server Help*. Give this new account a password and ensure that it is set to never expire.

2. When you are back in the tree view of the users in the directory, right-click the account and select **Reset Password**. Reset the password to the exact same password that you initially set the account to.

   **Note:** Resetting the password effectively clears the cache in the Active Directory and lets you proceed to the next step.

3. Type the information in the (now running) LiveCycle Policy Server configuration page within the domain setup. (See "Adding Kerberos authentication using the administration interface" on page 29 and the *LiveCycle Policy Server Help*.)

## Adding Kerberos authentication using the administration interface

When you select Kerberos from within the LiveCycle Policy Server administration interface, you get some fields to fill with configuration information. These fields are described in the *LiveCycle Policy Server Help*; however, this section provides some additional information.

The following table provides brief descriptions of the fields.

| Parameter | Description |
|---|---|
| DNS IP | The IP address of the DNS server such that it can refer to the Active Directory server by name (for example, 192.168.1.1). The DNS is needed to resolve the KDC Host. |
| KDC Host | The DNS name of the Active Directory server (for example, test.2003.policyserver.net). In Kerberos terminology, this is the Key Distribution Center (KDC) host. |
| Service User | The login name of the user of the special Active Directory account (for example, PolServerKerberos). |
| Service Realm | The Active Directory domain. This domain must be typed in capital letters (for example, 2003.POLICYSERVER.NET). |
| Service Password | The password for the special Active Directory account. |

The LiveCycle Policy Server client software in Acrobat 7.0 needs to know the Kerberos service that LiveCycle Policy Server is using to connect LiveCycle Policy Server to Active Directory. This service corresponds to the new account that is created in the steps presented in "Creating the Active Directory account" on page 28 and described in the KDC Host, Service User, Service Password, and Service Realm fields.

Technically, Acrobat requires the service in a special format constructed using the service principal (Service User) as well as the realm (Service Realm). How these fields are formatted is important. Simply add the simple user name (Service User) and password (Service Password). The realm (Service Realm) is the Active Directory domain name. Note that the realm must be typed in all uppercase letters.

**Tip:** If you are very familiar with Kerberos, it is possible to instead type a correctly created Service Principal Name (SPN) rather than the user name in the Service User field.

### Configuring support for Novell's eDirectory directory server

Novell's eDirectory directory server is supported as an LDAP server by LiveCycle Policy Server. To configure eDirectory as an authentication provider, install LiveCycle Policy Server to use LDAP configuration as described in this guide for your application server. Then, when you log into LiveCycle Policy Server as an administrator, use the following process.

➤ **To configure eDirectory as an authentication provider:**

1. In the eDirectory Help, select **Configuring User Authentication** > **Configuring domains** > **Adding domains** and follow the procedure "To add a domain."

2. When you add a new directory and populate the New Directory fields, first select the option to populate the page with "No defaults", and then proceed according to the parameters of your environment.

3. Be sure to use the value `edirectory` for Name Attribute for both users and groups. Do *not* use `entrydn`.

Depending on the application server you are using, configuration steps may be required from the administrative console to use eDirectory. For more information, see your application server's documentation.

## LiveCycle Reader Extensions

After deploying LiveCycle Reader Extensions to the application server, you can perform the following tasks:

- Create applications for LiveCycle Reader Extensions. (See the LiveCycle Reader Extensions *Developer's Guide*.)
- If you plan to use LiveCycle Reader Extensions to add usage rights to individual documents, you can use the LiveCycle Reader Extensions web application or an internal application (such as an EJB or servlet). (See "Accessing the LiveCycle Reader Extensions web application" on page 30.)

### Accessing the LiveCycle Reader Extensions web application

If you plan to use LiveCycle Reader Extensions to add usage rights to individual documents, you can use the LiveCycle Reader Extensions web application or an internal application (such as an EJB or servlet).

You must ensure that those responsible for adding usage rights know where the web application or internal application are located on your network. For information about setting usage rights, see the LiveCycle Reader Extensions *Developer's Guide* or *Web Application Help*.

➤ **To access the LiveCycle Reader Extensions web application:**

1. Ensure that LiveCycle Reader Extensions is deployed to the application server and the application server is running.

2. Go to the URL `http://<hostname>:8080/ReaderExtensions`.

   Type the username and password you created when you configured the LiveCycle Reader Extensions user. (See "Setting up users, roles, and login files" on page 41.)

   For turnkey installations, the default username is administrator and the default password is password. To edit these values, see "Setting up users, roles, and login files" on page 41.)

# LiveCycle Document Security

After deploying LiveCycle Document Security to the application server, you can perform the following tasks:

- Test the installation and deployment of LiveCycle Document Security by building and deploying the application samples that are available with the product. The samples demonstrate the different types of custom applications you can create using the LiveCycle Document Security APIs. For more information, see the Samples documentation located in the *[LiveCycle root]*/DocumentSecurity/samples/docs directory.
- Create applications for LiveCycle Document Security. (See the LiveCycle Document Security *Developer's Guide*.)

# Part II: Manual Configuration and Deployment

This section of the guide describes how to manually configure and deploy your LiveCycle products.

For information on the turnkey configuration and deployment of the products, see "Installing LiveCycle Products Using the Turnkey Installation" on page 20.

# 4 Installing LiveCycle Products

***This chapter applies to all LiveCycle security products.***

This chapter describes how to use the installation program to install LiveCycle Document Security, LiveCycle Policy Server, or LiveCycle Reader Extensions on a Windows or Linux operating system.

Before you install the product, you must ensure that your environment includes the software and hardware required to run LiveCycle products. You should also understand the installation options and have prepared the environment as required. (See "Before You Install" on page 9.)

If you are installing to a location where a LiveCycle product is already installed, install the new product, run Configuration Manager to reassemble the products, undeploy the deployed product from the application server, and then redeploy the files to the application server.

To install using the turnkey method, see "Installing LiveCycle Products Using the Turnkey Installation" on page 20.

## Performing the installation

When you run an installation program, you need the following information:

- The serial number for the product you are installing.
- The type of installation and configuration you are performing. (See "Methods for installing, configuring, and deploying LiveCycle products" on page 10.)

When installing on Linux, the installation program uses the logged-in user's home directory as a temporary directory for storing files. As a result, messages such as the following may appear in the console:

```
WARNING: could not delete temporary file /home/<username>/ismp001/1556006
```

When you complete the installation, you must manually delete the temporary files.

**Caution:** Ensure that the temporary directory for the operating system that you are using has a minimum of 800 MB of free space available for the installation program:

- (Windows) TMP or TEMP path as set in the Environment Variables
- (Linux) Logged-in user's home directory

**Tip:** To improve the speed of the installation, disable any on-access virus scanning software for the duration of the installation.

➤ **To install LiveCycle products for manual deployment:**

1. At the root level of the installation media, start the installation program:
   - (Windows) Double-click th exe file.
   - (Linux) From a command prompt, type: `file_name.bin`

**Caution:** When you are installing the product on a Linux operating system, you must be logged in as the root user to successfully install the product to the default location, /opt/adobe/livecycle/. If you are logged in as a non-root user, you must change the installation directory to one for which you have permissions (for example, $HOME/adobe/livecycle).

**Note:** You may have to change the file permissions for the Linux installation program. To do so, type: `chmod +x filename.bin`

2. On the Welcome screen, click **Next**.
3. Review the information on the Upgrading screen, and then click **Next**.
4. Type the serial number in the text box and click **Next**.
5. Read the Product License Agreement, select **I accept the terms of the license agreement**, and then click **Next**.
6. Select the application server you are deploying to and click **Next**.

   **Note:** For JBoss, do not select the option to automatically configure and deploy.

7. Accept the default directory as listed or click **Browse** and navigate to the directory where you want to install the product, and then click **Next**.

   **Note:** You can also enter a name for a new install directory and one will be created for you.

**Caution:** When you install the product, you can specify a different installation location. If you are installing on Linux the directory you specify cannot contain any spaces; otherwise, the installation program does not install the product.

8. Review the installation details, and then click **Install**. The installation program displays the progress of the installation. A summary screen appears when the product installation is completed.
9. (Windows) Ensure that the **Start Configuration Manager** option is not selected.
10. Click **Finish**.

**Note:** Do not run Configuration Manager until you have installed all of the products you require. After you have completed installing all of the products, you only need to run Configuration Manager once to configure and assemble the deployable components. You can also run Configuration Manager at another time to configure and assemble product modules. (See "Configuring LiveCycle Products" on page 44.)

# Viewing the error log

If any errors occur during the installation, the installation program creates a log file called log.txt, which contains the error messages. The log file is located in the *[LiveCycle root]* directory.

# Next steps

You must now prepare your database and application server for hosting LiveCycle products. (See "Preparing your Environment" on page 36.)

# 5 Preparing your Environment

***This chapter applies to all LiveCycle security products.***

This chapter describes how to prepare your environment for hosting LiveCycle products. You must perform the tasks provided in this chapter before you configure the application server:

- "Creating the LiveCycle database" on page 36
- "Preparing JBoss" on page 41
- "Installing database drivers" on page 43

You do not need to perform these tasks if you are performing a turnkey install.

This chapter assumes that you have installed your LiveCycle products. If you have not installed the products, see "Installing LiveCycle Products" on page 33.

## Creating the LiveCycle database

***This section applies to LiveCycle Policy Server only.***

This section describes how to set up the database that stores LiveCycle configuration information and run-time data. If you previously configured the database for deploying other LiveCycle products, you do not need to perform these tasks again.

If this is the first installation of a LiveCycle product, you need to create an empty database. All of the tables required to support LiveCycle products will be created by Configuration Manager when you initialize the database. (See "Initializing the Database" on page 63.)

If you are using the turnkey method to install the product, JBoss and MySQL are installed and configured automatically; you do not need to perform the tasks in this section.

If you stop and start the LiveCycle database while LiveCycle Policy Server is running, you also need to stop and start the LiveCycle Policy Server application. Depending on the application server LiveCycle Policy Server is running on, you may need to stop and start the application server as well. This establishes a new connection between LiveCycle Policy Server and the LiveCycle database.

### Creating a MySQL database

Use the MySQL tools to create a database for use with LiveCycle products and a MySQL user account that the application server can use to connect to the database. You also need to modify the MySQL database server configuration. For information about creating the database and user account, see the MySQL documentation.

You need the database name and the user name and password of the MySQL user account when you configure the database connection.

### MySQL user account

The MySQL user account that you create requires these privileges to access the tables in the LiveCycle database:

- SELECT
- INSERT
- UPDATE
- DELETE
- CREATE
- DROP
- REFERENCES
- INDEX
- ALTER
- CREATE_TMP_TABLE
- LOCK_TABLES

### MySQL database server configuration

To prevent issues from occurring when uploading large files to MySQL, you need to set the maximum allowed communication packet size to 25 MB. You can set this property in the MySQL my.ini file (Windows) or my.cnf file (Linux). For more information about the maximum allowed communication packet size in MySQL, go to http://dev.mysql.com/doc/mysql/en/packet-too-large.html.

If your MySQL installation does not include a my.ini or my.cnf file, you must create one. For information about the location of my.ini or my.cnf file, or how to create the file, see the MySQL documentation.

➤ **To customize the MySQL configuration:**

1. Open the my.ini file in a text editor.
2. Add the following line to the end of the my.ini file:

```
max_allowed_packet=25M
```

3. Save and close the my.ini file.

## Creating an Oracle database

If you prefer not to use the default database that was created when you installed Oracle 9i or 10g, create a new database using the Database Configuration Assistant tool.

If any of the LDAP directories that the new database will synchronize with for authenticating LiveCycle users includes records with UTF-8 characters, you must create a database that uses the UTF-8 character set.

You must also create a new user on the database and assign it the CONNECT and RESOURCE roles, as well as the ACCESS_ANY_WORKSPACE, UNLIMITED TABLESPACE, and CREATE VIEW system privileges. For deployments on Linux, the user name must not exceed 8 characters and, on Windows, it must not exceed 12 characters.

The user name and password of the new user you create on the database are used again when you create the data source.

The Connect role requires the following system privileges:

- Alter Session
- Create Cluster
- Create Database Link
- Create Sequence
- Create Session
- Create Synonym
- Create Table
- Create View

The Resource role requires the following system privileges:

- Create Cluster
- Create Index Type
- Create Operator
- Create Procedure
- Create Sequence
- Create Table
- Create Trigger
- Create Type

For information about using Oracle 9i or 10g, see the Oracle 9i or 10g user documentation.

## Creating a DB2 database

Create a DB2 database by running the script provided in this section. The script is tuned for a system that will use 1 GB of memory for the database. If your system has a different size of memory dedicated for the database, see the DB2 documentation for details on configuring your system settings.

You must also create a user with SYSADM and DBADM privileges that can be used when configuring the data source on the application server. For information about creating a user, see the DB2 documentation. For deployments on Linux, the user name must not exceed 8 characters and, on Windows, it must not exceed 12 characters.

The user name and password of the new user you create on the database are used again when you create the data source.

After you create the database, you need to configure it to enable concurrent usage. (See "Configuring a DB2 database for concurrent usage" on page 40.)

➤ **To create a DB2 database:**

1. On the computer that hosts DB2, create a new text file that includes the following DB2 script:

```
create database dbname using codeset utf-8 territory default;
connect to dbname;
CREATE BUFFERPOOL "BP8K" SIZE 50000 PAGESIZE 8192 NOT EXTENDED STORAGE;
connect reset;
connect to dbname;
CREATE TEMPORARY TABLESPACE DBNAME_TEMP_8K IN DATABASE PARTITION GROUP
IBMTEMPGROUP PAGESIZE 8192 MANAGED BY SYSTEM USING
('DB2_root\DBNAME_TEMP') EXTENTSIZE 32 PREFETCHSIZE 16 BUFFERPOOL BP8K;
CREATE REGULAR TABLESPACE DBNAME_DATA_8K IN DATABASE PARTITION GROUP
IBMDEFAULTGROUP PAGESIZE 8192 MANAGED BY DATABASE USING
(FILE'DB2_root\DBNAME_DATA'9000) EXTENTSIZE 16 PREFETCHSIZE 16 BUFFERPOOL
BP8K;
commit work;
connect reset;
connect to dbname;
alter bufferpool ibmdefaultbp immediate size 96000;
alter bufferpool bp8k immediate size 32000;
commit work;
connect reset;
update db cfg for dbname using dbheap 4000;
update db cfg for dbname using logbufsz 2048;
update db cfg for dbname using locklist 2000;
update db cfg for dbname using chngpgs_thresh 40;
update db cfg for dbname using logfilsiz 4000;
deactivate database dbname;
activate database dbname;
```

2. Make the following changes to the script:
   - Replace the instances of *dbname* and *DBNAME* with the name that you want for the LiveCycle database.
   - Replace *DB2_root* with the path to the *DBNAME*_TEMP and *DBNAME*_DATA files according to your DB2 installation.
   - Ensure that no commands include line breaks and each command is terminated by a semicolon (";").
   - Change 9000 in the following line based on your database size:

     ```
     (FILE'DB2_root\DBNAME_DATA'9000)
     ```

     This number specifies the minimum number of pages required to initialize the database. You can also change this number using the DB2 administration tools after initializing the database.

3. Save the text file in a location that DB2 Command Line Processor can access.

4. Open a command prompt.

5. (Windows) Enter the following command to open DB2 Command Line Processor:

   ```
   db2cmd
   ```

6. Enter the following command to run the script:

   ```
   db2 -tf <path_to_script_file>/<script_file_name>
   ```

### Configuring a DB2 database for concurrent usage

If you are using a DB2 database, you must configure it for multiple-user scenarios.

➤ **To configure the DB2 database for concurrent usage:**

1. Start DB2 Control Center:
   - (Windows) Click **Start** > **All Programs** > **IBM DB2** > **General Administration Tools** > **Control Center**.
   - (Linux) From a command prompt, enter the command: `db2cc`
2. In the DB2 Control Center object tree, click **All Databases.**
3. Right-click the database you created for LiveCycle products and click **Configuration Advisor**.
4. Follow the steps in the Configuration Advisor wizard and ensure that the properties in the following table are set.

## Creating a SQL Server database

Create a SQL Server database and create a user with DB_OWNER privileges that can be used when configuring the data source on the application server. For information about creating the database and user, see the SQL Server documentation.

The SQL Server database can be configured with Windows or SQL Server authentication types. For JBoss, the authentication type should be set to SQL Server.

You must download the Microsoft SQL Server JDBC drivers from the following website, and then copy them to the *[appserver root]*/lib directory:
www.microsoft.com/downloads/details.aspx?FamilyID=9f1874b6-f8e1-4bd6-947c-0fc5bf05bf71&DisplayLang=en.

For LiveCycle Policy Server, ensure that the following SQL Server properties are configured before creating the LiveCycle database:

- Set the **Minimum Memory Allocation** to the largest number your system can handle and ensure that **Dynamically configure SQL Server memory** is enabled.
- In processor properties of the database server, enable **Boost SQL Server priority on Windows** and **Use Windows NT fibers**.
- In database properties of the database server, set the **Recovery Interval** to 15 minutes.

After creating the database, ensure that the data and log files are set to a size large enough for your system. You must also set the growth size to a large number to ensure that data is not lost.

# Preparing JBoss

***This section applies to all LiveCycle security products.***

To prepare JBoss for the manual configuration of LiveCycle products, you must do the following:

- Install JBoss Application Server 3.2. 5. (See "Installing JBoss Application Server 3.2.5" on page 41.)
- Set up users, roles, and login files. (See "Setting up users, roles, and login files" on page 41.)
- Configure DocumentServicesLibrary.jar. (See "Configuring DocumentServicesLibrary.jar" on page 42.)
- Copy files. (See "Copying files" on page 42.)
- Copy files to run LiveCycle Policy Server with LiveCycle Workflow. (See "Copying files to run LiveCycle Policy Server with LiveCycle Workflow" on page 43.)

## Installing JBoss Application Server 3.2.5

To prepare JBoss for the manual configuration of LiveCycle products, you must download and install JBoss Application Server 3.2.5. You can obtain JBoss at this location:

http://labs.jboss.com/portal/jbossas/download

If you are performing a turnkey install, you do not need to install JBoss because Configuration Manager installs it automatically.

## Setting up users, roles, and login files

***This section applies to LiveCycle Reader Extensions only.***

To run LiveCycle Reader Extensions, you must set up a default user, user role, and user login. The installation program creates the appropriate files, but you must move them from the LiveCycle Reader Extensions installation directory to the application server deployment directory. After you have set up the default user, you can create additional users and user roles.

➤ **To set up default users, roles, and a login file:**

1. Copy the following files from the *[LiveCycle root]*/components/readerextensions/jboss/conf directory to the JBoss directory *[appserver root]*/server/all/conf directory:
   - ares-roles.properties
   - ares-users.properties
2. Stop and restart JBoss Application Server.

➤ **To create additional users and roles:**

1. In the *[appserver root]*/server/all/conf directory, open the text file named ares-users.properties.
2. Type the names of the users you want to create, and assign each user a password. For example:

```
Joe=23Kin89
Mary=FY9876
admin=adminpassword
```

3. Save the file.

4. Open the ares-roles.properties file.

5. In the file, type the name of each user you defined in the ares-users.properties file, and type the roles of each user. For example:

   ```
   Joe=AdobeReUser
   Mary=OtherRole
   admin=AdobeReUser
   ```

   The AdobeReUser is the user that has permission to access the LiveCycle Reader Extensions web application.

6. Save the file.

7. In the *[appserver root]*/server/all/conf directory, open the login-config.xml file and add the following text:

   ```
   <application-policy name = "ARES">
   <authentication>
   <login-module code = "org.jboss.security.auth.spi.UsersRolesLoginModule"
   flag = "required">
   <module-option name="usersProperties">ares-users.properties
   </module-option>
   <module-option name="rolesProperties">ares-roles.properties
   </module-option>
   </login-module>
   </authentication>
   </application-policy>
   ```

8. Save the login-config.xml file.

9. Stop and restart the JBoss application server.

## Configuring DocumentServicesLibrary.jar

***This section applies to LiveCycle Document Security and LiveCycle Reader Extensions only.***

You need to copy the DocumentServicesLibrary.jar file from the *[LiveCycle root]* directory to the *[appserver root]* directory.

➤ **To copy the DocumentServicesLibrary.jar file:**

- Copy the DocumentServicesLibrary.jar file from the *[LiveCycle root]*/components/csa/jboss/lib/adobe directory to the *[appserver root]*/server/all/lib directory.

## Copying files

***This section applies to LiveCycle Policy Server only.***

Before running Configuration Manager you must copy the following files:

- The jax-qname.jar file from the *[LiveCycle root]*/PolicyServer/sdk/lib/JBoss directory to the *[appserver root]*/server/all/lib directory.
- The edc-server-spi.jar file from the *[LiveCycle root]*/PolicyServer/sdk/spi-lib directory to the *[appserver root]*/server/all/lib directory.

## Copying files to run LiveCycle Policy Server with LiveCycle Workflow

***This section applies to LiveCycle Policy Server only.***

To run LiveCycle Policy Server with LiveCycle Workflow, you must copy the following library files from the *[LiveCycle root]*/PolicyServer/sdk/lib/JBoss directory to the *[LiveCycle root]*/components/workflow/ext directory:

- edc-sdk.jar
- jaxb-api.jar
- jaxb-impl.jar
- jaxb-libs.jar
- namespace.jar
- relaxngDatatype.jar
- sdk-ejb-client.jar
- xsdlib.jar

# Installing database drivers

***This section applies to LiveCycle Policy Server only.***

For LiveCycle Policy Server, you must install database drivers to the installation directories of the application server. Drivers are required to enable Configuration Manager and the application server to connect to the LiveCycle database. You need to install the drivers for the type of database that you use for the LiveCycle database.

➤ **To install the MS SQL Server 2000 SP 3 drivers:**

1. Download the MS SQL Server 2000 SP 3 database drivers from www.microsoft.com/downloads/details.aspx?familyid=07287B11-0502-461A-B138-2AA54BFDC03A&displaylang=en

2. Copy the msbase.jar, mssqlserver.jar, and msutil.jar files to the *[appserver root]*/lib directory.

➤ **To install the MySQL 4.1 driver:**

- Copy the mysql-connector-java-3.0.15-ga-bin.jar driver from the installation CD to the *[appserver root]*/lib directory.

# Next step

You must now configure the product for deployment using the manual configuration method. (See "Configuring LiveCycle Products" on page 44).

# 6 Configuring LiveCycle Products

***This chapter applies to all LiveCycle security products.***

This chapter describes how to configure LiveCycle products for deployment.

This chapter assumes that you have installed your LiveCycle products. If you have not installed the products, see "Installing LiveCycle Products" on page 33.

The instance of Configuration Manager that is installed with one LiveCycle product can be used with all LiveCycle products. When you run Configuration Manager, you can specify the LiveCycle products that you are configuring as well as the type of application server that you are leveraging in the solution. You can also set run-time properties for LiveCycle products and enable or disable security.

If you are already running one or more LiveCycle products, you must configure and assemble those existing products with the new one that you want to configure. To do this, you must undeploy the previously deployed products and select all of the products that you want to assemble during configuration. The previously deployed products are then included in the LiveCycle.ear file and deployed with the new product when you deploy the file to the application server.

**Note:** To assemble multiple LiveCycle products, each product must be installed in the same *[LiveCycle root]* directory.

When Configuration Manager completes the configuration of the products, it places the applicable files to be deployed to the application server (LiveCycle.ear, adobe-FontManager.ear, LiveCycle-security.ear, and edc-server.ear) in the following directory:

- (Windows) *[LiveCycle root]*\configurationManager\export
- (Linux) *[LiveCycle root]*/configurationManager/export

**Note:** Ensure that the application server is running before starting Configuration Manager. This applies to standalone and remote scenarios. Also ensure that Configuration Manager is running on the same computer that is hosting the application server. All configuration settings are final and cannot be reversed. There is not a cache of old or overwritten configuration settings.

If any errors occur during the configuration, Configuration Manager creates a log file called lcm.log, which may contain the error message. The log file is located in the *[LiveCycle root]*/configurationManager directory.

## Creating an endorsed directory

***This section applies to LiveCycle Policy Server only.***

Before running Configuration Manager for LiveCycle Policy Server, you must create an endorsed directory.

➤ **To create an endorsed directory:**

1. Navigate to the *[appserver root]*/server/all/lib directory and create a directory called *endorsed*.
2. Copy the following files from the *[LiveCycle root]*/components/Policyserver/endorsed directory to the endorsed directory you just created:
   - dom3-xercesImpl-2.4.0.jar
   - dom3-xml-apis-2.4.0.jar
   - xalan-2.4.1.jar

# Configuring LiveCycle products for deployment

➤ **To configure the products for deployment:**

1. Navigate to the *[LiveCycle root]*/configurationManager directory and start Configuration Manager:
   - (Windows) Double-click **ConfigurationManager.exe**.
   - (Linux) From a command prompt, type: `ConfigurationManager.bin`
2. If prompted, select a language for Configuration Manager and click **OK**.
3. On the Configuration Manager Welcome screen, click **Next**.
4. Select **Custom Configuration Wizard** and click **Next**.
5. If prompted, on the Configuration Preferences screen, select either **Use Previously Entered Values** or **Revert to Default Values**, and then click **Next**.
6. Select the application server you have installed, if displayed select **Foundation** and the products that you want to configure, and then click **Next**.
7. Ensure that only **Configure and assemble LiveCycle products** is selected, and then click **Next**.
8. On the Configure and Assemble Products screen, click **Next**.
9. (LiveCycle Policy Server) Proceed to step 27.
10. (LiveCycle Reader Extensions, LiveCycle Document Security) (Optional) On the Data Manager Module Configuration screen, if you are using SSL security on your application server, select **Enable SSL** and type the SSL credential password.

    If you have not yet set up your SSL credential, you can type a password here and use it when you create an SSL credential. For information about creating an SSL credential, see "Configuring SSL on JBoss" on page 65.
11. (LiveCycle Reader Extensions, LiveCycle Document Security) On the Data Manager Module Configuration screen, enter a directory to use for **Adobe LiveCycle products temp file**, and then click **Next**.

    For more information about the Adobe LiveCycle products temp file, see "Optimizing inline documents and impacts on JVM memory" on page 96.

12. (LiveCycle Reader Extensions, LiveCycle Document Security) On the Data Manager Module Configuration continued screen, accept the default values for the following properties or enter new values, and then click **Next**:
    - **Local storage sweep interval (in seconds):** The amount of time between attempts to delete any files that are no longer needed and were used to pass the document data between LiveCycle services running on the same computer.
    - **Global storage sweep interval (in seconds):** The amount of time between attempts to delete any obsolete files that were used to pass the document data between LiveCycle services running on different computers. Specify this property only when deploying LiveCycle products in a clustered environment.
    - **Default maximum inline size (in bytes):** The maximum number of bytes kept in memory when passing documents between different LiveCycle components. Documents that exceed this maximum are stored on the hard drive. Use this property for performance tuning. (See "Optimizing inline documents and impacts on JVM memory" on page 96.)
    - **Default disposal time-out (in seconds):** The maximum amount of time during which a document being passed between different LiveCycle components is considered active. After this time has passed, any files used to store this document are subject to removal. Use this property to control the usage of disk space.
    - **Use NFS protocol (Windows only):** Select this option when deploying LiveCycle products in a clustered environment. Additional NFS software should be installed on your computer running Windows before enabling this option. This option does not affect deployments on Linux.
    - **Global storage directory:** A path to a shared directory used to store long-lived documents that are passed between LiveCycle products. Using an NFS shared directory can help to improve performance. (See "Optimizing inline documents and impacts on JVM memory" on page 96.)

13. (LiveCycle Reader Extensions, LiveCycle Document Security) (Optional) On the Font Manager Module Configuration screen, select the fonts for the product to use in addition to the fonts that are included with the product. In the **Fonts directory** box, type the path or browse to the directory that contains the fonts to add, and then click **Next**.

    **Note:** Your right to use fonts provided by parties other than Adobe is governed by the license agreements provided to you by such parties in connection with those fonts, and is not covered under your license to use Adobe software. Adobe recommends that you review and ensure you are in compliance with all applicable non-Adobe license agreements before using non-Adobe fonts with Adobe software, particularly with respect to use of fonts in a server environment.

14. (LiveCycle Reader Extensions, LiveCycle Document Security) On the Trust Directory Selection screen, select one of the following options:
    - If you do not have a trust directory, select **Create a new trust directory** and click **Next**. Proceed to step 15.
    - To use an existing trust directory, select **Use existing trust directory**, click **Browse** to navigate to and select the directory, and then click **Next** and proceed to step 20. A valid existing trust directory must contain the following files and directories:
        - /trust.xml
        - /credentials/
        - /certificates/
        - /CRLs/
        - /keystore
        - /trust.sig

15. (LiveCycle Reader Extensions, LiveCycle Document Security) On the Java Home Selection screen, browse to the location of the application server JDK and click **Next**.

16. (LiveCycle Reader Extensions, LiveCycle Document Security) On the New Trust Directory Configuration screen, click **Browse** and navigate to the directories containing your CRLs, credentials, and certificates, and then click **Next**. If you do not want to include copies of your CRLs, credentials, and certificates in the trust directory, click **Next**.

17. (LiveCycle Reader Extensions, LiveCycle Document Security) On the Keystore Selection screen, select one of the following options:

    - If you do not have a keystore and pair, select **Create a new keystore and pair** and click **Next**. On the Keystore Creation screen, you must type a password, alias, and key password for the new keystore and pair and click **Next**. On the Keystore Creation - DN Values screen, complete the DN registration form and click **Next**.
    - To use an existing keystore and pair, select **Use an existing keystore and key pair** and click **Next**. On the Existing Keystore screen, type the password, alias, and key password for your keystore and key pair and click **Next**. On the Trust Manager Module Configuration screen, type additional keystore and signature information or, if you are not sure what these values should be, accept the defaults, and click **Next**.

18. (LiveCycle Reader Extensions) On the Reader Extensions Credential screen, browse to the location of your credential file, type the credential password, and then click **Next**.

19. (LiveCycle Reader Extensions, LiveCycle Document Security) On the Trust XML Review screen, review the contents of the trust.xml file and click **Next**.

    To edit the trust.xml file, open the trust.xml file from the *[product root]*/trust/ directory in a text editor, make your changes, save the file, and click **Next**. Changes to the trust.xml file are not displayed on the screen. (For more information on the file contents, see "Content and Format of the trust.xml File" on page 103.).

    For LiveCycle Reader Extensions, proceed to step 23. For LiveCycle Document Security, proceed to step 25.

    **Note:** By default, full privileges are granted to the imported certificates. You can edit these privileges before continuing the installation.

20. (LiveCycle Reader Extensions, LiveCycle Document Security) On the Existing Keystore screen, type a password, alias, and key password for your keystore and key pair, and then click **Next**.

    For LiveCycle Document Security, proceed to step 25.

21. (LiveCycle Reader Extensions, LiveCycle Document Security) On the Trust Manager Module Configuration screen, type additional keystore and signature information or, if you are not sure what these values should be, accept the defaults and click **Next**.

22. (LiveCycle Reader Extensions) On the Credential Alias Selection screen, type the credential alias as configured in the trust.xml file, type the credential password, and then click **Next**.

23. (LiveCycle Reader Extensions) On the Reader Extensions Configuration screen, complete the product configuration, and then click **Next**:
    - **Directory for PDF uploads and storage**: The root directory for product application files, including uploaded and result files. This directory can be specified as an absolute path or relative to the web application directory. The default value is /WEB-INF/work.
    - **Days before results are expired**: The expiration period (in days) for result files. Result files and uploaded files are deleted no sooner than the expiration directory. The default value is 3.
    - **Message to display if a later version of Adobe Reader is required**: The default message to show users of a rights-enabled PDF document if the document requires a later version of Adobe Reader.
    - **URL to download later version of Adobe Reader**: The URL where the users of the rights-enabled PDF documents can obtain the latest version of Adobe Reader.
24. (LiveCycle Reader Extensions) On the Reader Extensions Locale Configuration screen, configure the default locale settings for the product, and then click **Next**:
    - **Respect browser locale**: Specifies whether the web application user's browser uses the browser-specified locale. If the value is False, the value specified by the defaultISO639LanguageCode and defaultISO3166CountryCode parameters are used. The default value is True.
    - **Default language code**: The ISO 639 Language Code for the default language. The default value is en.
    - **Default country code:** The ISO 3166 Country Code for the default country. The default value is US.
25. (LiveCycle Reader Extensions, LiveCycle Document Security) On the PDFAgent Module Configuration screen, configure the maximum number of simultaneous processes allowed on the application server. If you do not want a limit on the number of simultaneous processes, set the value to 0. If you are not sure what the value should be, accept the default value of 2.
26. (LiveCycle Reader Extensions, LiveCycle Document Security) On the APSProxy Module Configuration screen, if applicable, verify that the LiveCycle Policy Server host name, RMI Port, and application server initial context factory values are correct, and then click **Next**.

    **Note:** LiveCycle Policy Server must be located on the same computer or subnet as LiveCycle Reader Extensions or LiveCycle Document Security.
27. (LiveCycle Policy Server) Select the default locale to ensure that audit events are recorded and displayed in the appropriate language, and then click **Next**.
28. Review the configuration summary information and click **Back** to change any settings or click **Next** to continue. When you continue, Configuration Manager configures the LiveCycle product properties using the specified values and assembles the products into the LiveCycle.ear file.
29. On the Configure and Assemble products summary screen, click **Next**.
30. Click **Finish** to close Configuration Manager.

## Next step

You can now manually configure the application server. (See "Manually Configuring JBoss" on page 49.)

# 7 Manually Configuring JBoss

This chapter describes how to manually configure JBoss Application Server to prepare for the manual deployment of LiveCycle Policy Server, LiveCycle Reader Extensions, and LiveCycle Document Security.

If you installed the product using the turnkey method, you only need to configure JAAS authentication. (See "Configuring JAAS authentication" on page 55.)

For LiveCycle Policy Server, this chapter assumes that you have configured your database and created a database user account. For LiveCycle Reader Extensions and LiveCycle Document Security, this chapter assumes that you have prepared your application server. (See "Preparing your Environment" on page 36.)

You need to perform the following tasks:

- Set up the JBoss environment. (See "Setting up JBoss" on page 50.)
- (LiveCycle Policy Server) Copy LiveCycle Policy Server product files to the JBoss directories. (See "Copying the JBoss configuration files" on page 50.)
- Configure JBoss properties. (See "Configuring JBoss properties" on page 51.)
- Configure the transaction time-out value. (See "Configuring the transaction time-out property" on page 51.)
- (LiveCycle Policy Server) Set up the connection to the LiveCycle database. (See "Connecting JBoss to the database" on page 52.)
- (LiveCycle Policy Server) Set up authentication. (See "Configuring security" on page 55.)

This chapter refers to the location where JBoss Application Server is installed as *[appserver root]* and the location where LiveCycle products and components are installed as *[LiveCycle root]*.

## Starting and stopping JBoss

Several procedures in this chapter require you to stop and start the instance of JBoss where you want to deploy LiveCycle products.

➤ **To start JBoss:**

1. From a command prompt, navigate to the *[appserver root]*/bin directory.
2. Start the server by entering the following command:
   - (Windows) `jboss-run.bat -c all`
   - (Linux) `./jboss-run.sh -c all`

➤ **To stop JBoss:**

1. From a command prompt, navigate to the *[appserver root]*/bin directory.
2. Stop the server by entering the following command:
   - (Windows) `shutdown.bat -S`
   - (Linux) `./shutdown.sh –S`

# Setting up JBoss

Before configuring JBoss and deploying LiveCycle products, you must set up the JBoss environment by completing the following tasks:

- Install JDK1.4.2_04 or higher.
- The JAVA_HOME variable should be set to the java home/jdk used for the application server.
- The PATH variable should include JAVA_HOME/bin as the fist entry.
- Install JBoss 3.2.5. The JBoss product files are located in the *[LiveCycle root]*/components directory.
- Define the JBOSS_HOME environment variable and set it to the JBoss installation directory.
- Stop JBoss if it is currently running. (See "Starting and stopping JBoss" on page 49.)

# Copying the JBoss configuration files

***This section applies to LiveCycle Policy Server only.***

LiveCycle Policy Server provides several files that you need to copy to the JBoss configuration directories. These files need to be copied on the instance of JBoss that hosts LiveCycle Policy Server.

Before copying the files, ensure that you make backup copies of all of the files. If a file exists in the target location, replace it with the new file.

➤ **To copy the JBoss configuration files:**

1. Copy the files listed in the table below to the target location.

| File | Source location | Target location |
|---|---|---|
| run.sh (Linux) or run.bat(Windows) | *[appserver root]*/PolicyServer/jboss-conf/scripts | *[appserver root]*/bin |
| log4j.xml | *[LiveCycle root]*/PolicyServer/jboss-conf/logging | *[appserver root]*/server/all/conf |
| server.xml | *[LiveCycle root]*/PolicyServer/jboss-conf/ssl | *[appserver root]*/server/all/deploy/jbossweb-tomcat50.sar |
| aps.keystore | *[LiveCycle root]*/PolicyServer/jboss-conf/ssl | *[appserver root]*/server/all/conf |
| XML files | *[LiveCycle root]*/PolicyServer/jboss-conf/jms<br>Copy the appropriate MySQL, Oracle, or MSSQL XML files and the edc-destinations-service.xml file. | *[appserver root]*/server/all/deploy-hasingleton/jms |
| axis.jar | *[LiveCycle root]*/PolicyServer/sdk/lib/JBoss | *[appserver root]*/server/all/deploy/jboss-net.sar |

2. Remove the hsqldb-jdbc2-service.xml fle from the *[appserver root]*/deploy-hasingleton/jms directory.

3. Open the jboss-service.xml file (*[appserver root]*/server/all/conf) in a text editor and after the following text:

   ```
   <!-- EAR deployer, remove if you are not using Web layers -->
   <mbean code="org.jboss.deployment.EARDeployer"
   name="jboss.j2ee:service=EARDeployer">
   ```

   Add the following two lines:

   ```
   <attribute name="Isolated">true</attribute>
   <attribute name="CallByValue">true</attribute>
   ```

   Save the file.

4. Open the jboss-service.xml file (*[appserver root]*/server/all/deploy/jbossweb-tomcat55.sar/META-INF) and change the `UseJBossWebLoader` attribute from true to false:

   ```
   <attribute name="UseJBossWebLoader">false</attribute>
   ```

   Save the file.

# Configuring JBoss properties

***This section applies to LiveCycle Reader Extensions and LiveCycle Document Security only.***

You must edit the `jacorb.poa.thread_pool_max` property in the jacorb.properties file to ensure that your Adobe document services run properly on the application server. The jacorb.properties file is located in the *[appserver root]*/server/all/conf directory.

Before making any changes, ensure that you create a backup copy of the jacorb.properties file.

➤ **To edit the jacorb.properties file:**

1. Open the jacorb.properties file in a text editor.
2. Locate the `jacorb.poa.thread_pool_max` property and change the property value to 16.

# Configuring the transaction time-out property

***This section applies to LiveCycle Reader Extensions and LiveCycle Document Security only.***

You must modify the transaction time-out value for your JBoss Application Server to prevent transaction time-out exceptions.

➤ **To configure the transaction time-out property:**

1. Navigate to the *[appserver root]*/server/all/conf directory and open the jboss-service.xml file in a text editor.
2. Set the value for `<attribute name="TransactionTimeout"></attribute>` to a higher number. For example, if the value is 300, set it to 300000 or 3000000. This value is in seconds.
3. Save the file and restart the application server.

# Connecting JBoss to the database

***This section applies to LiveCycle Policy Server on***ly.

You must configure the data source to connect the instance of JBoss that hosts LiveCycle Policy Server to the LiveCycle database. For JBoss, you can use a MySQL, SQL Server, Oracle, or DB2 data source.

## Configuring a MySQL data source

To enable LiveCycle Policy Server to communicate with a MySQL database that stores LiveCycle data, you must create a data source file and deploy it to the instance of JBoss that hosts LiveCycle Policy Server.

You need to create the data source file only if you want to manually deploy LiveCycle products. When you perform a turnkey installation, the application server and product are automatically configured to interact with the MySQL database, which is also automatically installed.

➤ **To create the data source file:**

1. Create an XML file using the following code:

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
   <local-tx-datasource>
      <jndi-name>EDC_DS</jndi-name>
      <connection-url>jdbc:mysql://localhost:3306/adobe</connection-url>
      <driver-class>com.mysql.jdbc.Driver</driver-class>
      <user-name>adobe</user-name>
      <password>adobe</password>
      <min-pool-size>1</min-pool-size>
      <max-pool-size>100</max-pool-size>
      <blocking-timeout-millis>20000</blocking-timeout-millis>
      <idle-timeout-minutes>10</idle-timeout-minutes>
      <prepared-statement-cache-size>50</prepared-statement-cache-size>
      <transaction-isolation>TRANSACTION_READ_COMMITTED
      </transaction-isolation>
   </local-tx-datasource>
</datasources>
```

2. Replace the bold text for the following elements with values that are specific to your LiveCycle database:
   - `<connection-url>` describes the computer name (localhost, or you can use the computer name, IP address, or fully-qualified path) and database port number, and the database name. The application server uses the URL to connect to the database.
   - `<user-name>` and `<password>` are the user name and password that the application server uses to access the database. These values are set when you create the database. (See "Creating a MySQL database" on page 36.)
3. Save the file as adobe-ds.xml in the *[appserver root]*/server/all/deploy directory.
4. Restart JBoss.

## Configuring a SQL Server data source

To enable LiveCycle Policy Server to communicate with a SQL Server database that stores LiveCycle data, you must create a data source file and deploy it to the instance of JBoss that hosts LiveCycle Policy Server.

For information about using SQL Server, see the SQL Server documentation.

➤ **To create the data source file:**

1. Create an XML file using the following code:

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
   <local-tx-datasource>
      <jndi-name>EDC_DS</jndi-name>
      <connection-url>jdbc:microsoft:sqlserver://localhost:1433;
      DatabaseName=adobe;SelectMethod=Cursor</connection-url>
      <driver-class>com.microsoft.jdbc.sqlserver.SQLServerDriver
      </driver-class>
      <user-name>adobe</user-name>
      <password>adobe</password>
      <SelectMethod>Cursor</SelectMethod>
      <min-pool-size>1</min-pool-size>
      <max-pool-size>100</max-pool-size>
      <blocking-timeout-millis>20000</blocking-timeout-millis>
      <idle-timeout-minutes>10</idle-timeout-minutes>
      <check-valid-connection-sql>SELECT 1</check-valid-connection-sql>
   </local-tx-datasource>
</datasources>
```

2. Replace the bold text for the following elements with values that are specific to your LiveCycle database:
   - `<connection-url>` describes the computer name (localhost, or you can use the computer name, IP address, or fully-qualified path) and port number, and the database name. The application server uses the URL to connect to the database.
   - `<user-name>` and `<password>` are the user name and password that the application server uses to access the database. These values are set when you create the database. (See "Creating a SQL Server database" on page 40.)
3. Save the file as adobe-ds.xml in the *[appserver root]*/server/all/deploy directory.
4. Restart JBoss.

## Configuring an Oracle data source

To enable LiveCycle Policy Server to communicate with an Oracle tablespace that stores LiveCycle products run-time data, you must create an Oracle data source on JBoss and deploy it to the instance of JBoss that hosts LiveCycle Policy Server.

For information about the Oracle connection URL, see www.oracle.com/technology/tech/java/sqlj_jdbc/htdocs/jdbc_faq.htm#05_04.

➤ **To create the data source file:**

1. Create an XML file using the following code:

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
  <local-tx-datasource>
    <jndi-name>EDC_DS</jndi-name>
    <connection-url>jdbc:oracle:thin:@host_name:port:database_name
    </connection-url>
    <driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
    <user-name>database_username</user-name>
    <password>password</password>
    <!-- Checks the Oracle error codes and messages for fatal errors -->
    <exception-sorter-class-name>
    org.jboss.resource.adapter.jdbc.vendor.OracleExceptionSorter
    </exception-sorter-class-name>
  </local-tx-datasource>
</datasources>
```

2. Replace the bold text for the following elements with values that are specific to your LiveCycle database:
   - `<connection-url>` describes the computer name (localhost, or you can use the computer name, IP address, or fully-qualified path) and port number, and the database name. The application server uses the URL to connect to the database.
   - `<user-name>` and `<password>` are the user name and password that the application server uses to access the database. These values are set when you create the database. (See "Creating an Oracle database" on page 37.)
3. Save the file as adobe-ds.xml in the *{appserver root]*/server/all/deploy directory.
4. Restart JBoss.

## Configuring a DB2 data source

To enable JBoss to communicate with a DB2 database that stores LiveCycle data, you must create a DB2 data source on JBoss and deploy it to the instance of JBoss that hosts LiveCycle Policy Server.

➤ **To create the data source file:**

1. Create an XML file using the following code:

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
  <local-tx-datasource>
    <jndi-name>EDC_DS</jndi-name>
    <connection-url>jdbc:db2://<server_name>:<port>/<database_name>
    </connection-url>
    <driver-class>com.ibm.db2.jcc.DB2Driver
    </driver-class>
    <user-name>adobe</user-name>
    <password>adobe</password>
    <SelectMethod>Cursor</SelectMethod>
    <min-pool-size>1</min-pool-size>
    <max-pool-size>100</max-pool-size>
```

```
      <blocking-timeout-millis>20000</blocking-timeout-millis>
      <idle-timeout-minutes>10</idle-timeout-minutes>
    </local-tx-datasource>
</datasources>
```

2. Replace the bold text for the following elements with values that are specific to your LiveCycle database:
   - `<connection-url>` describes the server name (the name of the computer that hosts DB2) and port number, and the database name. The application server uses the URL to connect to the database.
   - `<user-name>` and `<password>` are the user name and password that the application server uses to access the database. These values are set when you create the database. (See "Creating a DB2 database" on page 38.)
3. Save the file as adobe-ds.xml in the *[appserver root]*/server/all/deploy directory.
4. Restart JBoss.

# Configuring security

***This applies to LiveCycle Policy Server only.***

Administrators and users must be authenticated to access LiveCycle Policy Server features. LiveCycle Policy Server supports form-based authentication and basic authentication, by default forms-based JAAS authentication is used.

To implement basic authentication, you use the LiveCycle Policy Server SDK to integrate a custom authentication service provider. For more information, see the LiveCycle Policy Server *Developing Custom Applications* guide or contact Adobe Customer Support.

**Note:** LDAP exchanges information in clear text. You should configure JBoss to send authentication information over an SSL connection. (See "Configuring SSL on JBoss" on page 65.)

## Configuring JAAS authentication

For both turnkey and manual installations of LiveCycle Policy Server on JBoss, you need to configure JAAS authentication.

To configure JAAS authentication, you must add an application policy element in the login-config.xml file that JBoss uses so that LiveCycle Policy Server can communicate with your LDAP server. LiveCycle Policy Server provides a default login-config.xml file that includes the sample application policy element `<application-policy name="UsernamePwd_Auth_Search">`.

For information about the sample application policy element, see "Sample application policy" on page 56. For information about configuring LiveCycle Policy Server to use the application policy for authentication, see "LiveCycle Policy Server" on page 26.

➤ **To create the application policy for turnkey installations:**

1. Open the login-config.xml file from the *[LiveCycle root]*/jboss-3.2.5/server/all/conf directory.
2. Add and modify the application policy node with the properties of your LDAP directory.
3. Save the file.
4. Restart the application server.

➤ **To create the application policy for manual installations:**

1. Open the login-config.xml file from the *[appserver root]*/server/all/conf directory:
   - If you are deploying LiveCycle Policy Server to an existing instance of JBoss that is already running other LiveCycle applications, you can copy the sample application policy element from the default login-config.xml file, insert it into the login-config.xml file that is in the *[appserver root]*/conf directory, and then modify the application policy element appropriately.
   - If you are deploying LiveCycle Policy Server to a new instance of JBoss that is not running other LiveCycle applications, replace the login-config.xml file located in the *[appserver root]*/conf directory with the default login-config.xml file that LiveCycle Policy Server provides, and then modify the application policy element appropriately.
2. Add and modify the application policy node with the properties of your LDAP directory.
3. Save the file.
4. Restart the application server.

## Sample application policy

LiveCycle Policy Server provides a sample application policy to use as a template for setting up JAAS authentication on JBoss. The sample application policy is appropriate when the DN of the user records in your LDAP does not include the user's login identification.

To locate a user record, you configure a search filter that searches the object classes that include the user's login identification. For the search filter, you specify the prefix and the suffix to the user identification.

For example, if your directory uses the schema attribute `uid` for the user login identification, you would use the following XML for the search filter option:

```
<module-option name =
"searchfilterPrefix">(&amp;(objectClass=*)(uid=</module-option>
<module-option name = "searchfilterSuffix">))</module-option>
```

When a user logs in, LiveCycle Policy Server retrieves the user identification and uses it to build the search filter. It then searches for the user record to use for authentication.

**Tip:** You need to escape special characters in your XML code.

The sample application element is named `UsernamePwd_Auth_Search`, and resides in the login-config.xml file in the *[LiveCycle root]*/product/conf directory. You can copy the element to the login-config.xml file in the *[appserver root]*/conf directory.

**Note:** The Internet Engineering Task Force (IETF) Request for Comments (RFC) 2254 defines the syntax for LDAP search filters. For information about search filters that is specific to the LDAP server, see the documentation for the LDAP server.

For example application policies for Sun ONE and Active Directory, see "Example application policies" on page 58.

The following table describes each property that can be configured.

| Configurable option | Description |
|---|---|
| user.provider.url | The LDAP URL to your directory server (for example, ldap://servername:port) |
| java.naming.security.authentication | The LDAP authentication type. |
| searchUser | Set to true to force a search for the user with the dynamically constructed DN. |
| searchUsingAnonymousBind | Specifies whether access to the directory is controlled by using authentication:<br>• `true` - No authentication is performed, and no user information is needed to perform the search.<br>• `false` - Authentication is performed. A user identification and password are required to perform the search. |
| binduser | The DN of the user record that can access the directory for searching. This option has no value when searchUsingAnonymousBind is `true`. |
| bindpassword | The password associated with the DN specified in the binduser module option. This option has no value when searchUsingAnonymousBind is `true`. |
| basedn | The base DN of your directory. |
| searchfilterprefix | The portion of the search filter on the left of the user identification. |
| searchfiltersuffix | The portion of the search filter on the right of the user identification. |

## Example application policies

The following two application policy nodes are examples for use with Sun ONE and Active Directory LDAP servers. The examples use the hypothetical company *company_name.com* on the LDAP computer named *XYZ*.

### Example 7.1 *Sun ONE application policy node*

```
<application-policy name="UsernamePwd_Sun ONE">
  <authentication>
    <!-- do not change the following two lines-->
    <login-module
    code="com.adobe.edc.server.provider.authentication.login.LDAPLogin
    Module" flag="required"><module-option name =
    "java.naming.factory.initial">com.sun.jndi.ldap.LdapCtxFactory
    </module-option>

    <!-- this should be an LDAP url with server name and port-->
    <module-option name =
    "user.provider.url">ldap://xyz:389</module-option>

    <!-- this is the ldap authentication type.-->
    <module-option name =
    "java.naming.security.authentication">simple</module-option>

    <!-- setting this to true forces the code to search for the user with
    the DN that will be constructed dynamically.-->
    <module-option name = "searchUser">true</module-option>

    <!-- if searchUser is true then than the following three configure
    whether the search is performed anonymously or with a specific user-->
    <module-option name = "searchUsingAnonymousBind">true</module-option>
    <module-option name = "binduser"></module-option>
    <module-option name = "bindpassword"></module-option>

    <!-- this specifies what the basedn for users should be. Be sure this
    matches the directory settings that you specify in the Policy Server
    web pages or the end user will not successfully authenticate-->
    <module-option name =
    "basedn">ou=users,dc=company_name,dc=com</module-option>
    <module-option name =
    "searchfilterPrefix">(&amp;(objectClass=*)(uid=</module-option>
    <module-option name =
    "searchfilterSuffix">))</module-option></login-module>

  </authentication>
</application-policy>
```

**Example 7.2** ***Active Directory application policy node***

```
<application-policy name="UsernamePwd_ADS">
<authentication>

  <!-- do not change the following two lines-->
  <login-module code="com.adobe.edc.server.provider.
  authentication.login.LDAPLoginModule" flag="required">
    <module-option name = "java.naming.factory.initial">com.sun.jndi.
    ldap.LdapCtxFactory</module-option>

    <!-- this should be an LDAP url with server name and port-->
    <module-option name =
    "user.provider.url">ldap://xyz:389</module-option>

    <!-- this is the ldap authentication type.-->
    <module-option name =
    "java.naming.security.authentication">simple</module-option>

    <!-- setting this to true forces the code to search for the user with
    the DN that will be constructed dynamically.-->
    <module-option name = "searchUser">true</module-option>

    <!-- if searchUser is true then than the following three configure
    whether the search is performed anonymously or with a specific user-->
    <module-option name = "searchUsingAnonymousBind">false</module-option>
    <module-option name = "binduser">cn=John
    Doe,cn=users,dc=company_name,dc=com</module-option><module-option
    name = "bindpassword">password</module-option>

    <!-- this specifies what the basedn for users should be. Be sure this
    matches up with the directory settings in the config ui or else the end
    user will not successfully authenticate-->
    <module-option name =
    "basedn">cn=users,dc=company_name,dc=com</module-option>
    <module-option name = "searchfilterPrefix">(&amp;(objectClass=*)
    (sAMAccountName=</module-option>
    <module-option name = "searchfilterSuffix">)
    (!(userAccountControl:1.2.840.113556.1.4.803:=2)))</module-option>
  </login-module>
</authentication>
</application-policy>
```

# Next step

You can now deploy the product to the application server database. (See "Manually Deploying to JBoss" on page 60.)

# 8 Manually Deploying to JBoss

***This chapter applies to all LiveCycle security products.***

This chapter describes how to deploy your LiveCycle products to JBoss:


- "About deploying LiveCycle products" on page 60
- "Deploying to JBoss" on page 61
- "Resetting the counter for LiveCycle Reader Extensions" on page 62
- "Viewing log information" on page 62


This chapter assumes that you have configured the application server. If you have not done this, see "Manually Configuring JBoss" on page 49.

## About deploying LiveCycle products

Before you deploy LiveCycle products, ensure that you have the required software and files installed and know the location of the directories you will be working with. (See "System requirements" on page 11.)

After you have deployed the products, if you need to make any further changes to the run-time properties set during configuration, you can run Configuration Manager to make the changes, and then redeploy the updated EAR file. (See "Configuring LiveCycle Products" on page 44.)

If you are using an external web server, see your web server documentation for information on the configuration required to allow access to the application server.

If you have not yet configured optional security features such as IIOP over SSL on your application server, perform this configuration after you ensure that you have deployed your product modules successfully. (See "Configuring SSL on JBoss" on page 65.)

### JBoss directory name

This chapter refers to the location where JBoss Application Server is installed as *[appserver root]*, and the location where LiveCycle products and components are installed as *[LiveCycle root]*. (See "Conventions used in this guide" on page 6.)

## Summary of deployable components

This table lists the deployable components for LiveCycle security products.

| Component | LiveCycle product |
|---|---|
| adobe-FontManager.ear | LiveCycle Reader Extensions<br>LiveCycle Document Security |
| LCM.ear | All |
| LiveCycle.ear | LiveCycle Reader Extensions<br>LiveCycle Document Security |
| LiveCycle-security.ear | LiveCycle Reader Extensions<br>LiveCycle Document Security |
| edc-server.ear | LiveCycle Policy Server |

# Deploying to JBoss

You deploy LiveCycle products to JBoss by copying the deployable components to the JBoss deployment directory. JBoss can be running or stopped when you copy the files to the directory. After you copy the files, you must start or restart the server to ensure that the services start correctly.

➤ **To deploy LiveCycle products to JBoss:**

1. Copy the following files from the *[LiveCycle root]*/configurationManager/export directory to the *[appserver root]*/server/all/deploy directory:
   - (LiveCycle Reader Extensions, LiveCycle Document Security) adobe-FontManager.ear
   - (LiveCycle Reader Extensions, LiveCycle Document Security) LiveCycle.ear
   - (LiveCycle Reader Extensions, LiveCycle Document Security) LiveCycle-security.ear
   - (LiveCycle Policy Server) edc-server.ear
2. Copy the LCM.ear file from the *[LiveCycle root]*/configurationManager/deploy/jboss directory to the *[appserver root]*/server/all/deploy directory.
3. Restart JBoss to ensure that the applications start up.

   **Note:** When you start JBoss, you may see several error messages in the JBoss server log file. These messages occur if you have not yet initialized the database for LiveCycle Policy Server.

# Resetting the counter for LiveCycle Reader Extensions

You can optionally reset the counter for LiveCycle Reader Extensions.

➤ **To reset the counter:**

1. Stop the application server.
2. Delete the count.dat file from the *[LiveCycle root]*/jboss/server/all/tmp/deploy/tmp9731LiveCycle-security.ear-contents/ares.war/WEB-INF/work directory.

# Viewing log information

Events, such as run-time or startup errors, are recorded to the application server log files. If you have any problems deploying to the application server, you can use log files to help you find the problem. You can open the log files using any text editor.

The following log files are located in the *[appserver root]*/server/all/log directory:

- boot.log
- (LiveCycle Document Security) server.log. *yyyy-mm-dd*
- (LiveCycle Reader Extensions) server.log. *yyyy-mm-dd*
- server.log

# Next step

For LiveCycle Policy Server, you must now initialize the database using Configuration Manager. (See "Initializing the Database" on page 63.)

For LiveCycle Reader Extensions and LiveCycle Document Security, you can now run samples, finalize the product configuration, and develop applications. (See "Post-deployment" on page 26.)

# 9 Initializing the Database

***This chapter applies to LiveCycle Policy Server only.***

This chapter describes how to initialize the LiveCycle database for LiveCycle Policy Server.

This chapter assumes that you have configured the LiveCycle database for integration with the product and deployed the product to the application server. If you have not done this, see "Manually Deploying to JBoss" on page 60 and "Preparing your Environment" on page 36.

Initializing the database is a process that prepares a database for use with LiveCycle products, by creating tables and optionally adding data to them. You only need to perform this task the first time you deploy the product. When you run Configuration Manager to initialize the database, the application server must be running.

➤ **To initialize the database:**

1. Start the application server.
2. Start Configuration Manager by navigating to the *[LiveCycle root]*/configurationManager directory and entering the following command:
   - (Windows) `ConfigurationManager.exe`
   - (Linux) `ConfigurationManager.bin`
3. On the Welcome screen, click **Next**.
4. Select **Custom Configuration Wizard** and click **Next**.
5. If prompted, on the Configuration Preferences screen, select either **Use Previously Entered Values** or **Revert to Default Values**, and then click **Next**.
6. Select the type of application server that you are using.
7. If displayed, select **Foundation**, select the products that you want to configure and deploy, and then click **Next**.
8. Select **Bootstrap database** and click **Next**.
9. On the Bootstrap Database screen, click **Next**.
10. Click **Initialize Database Now**.
11. When initialization is complete, click **Next**, and then click **Finish**.

## Next step

You can now configure LiveCycle Policy Server run-time properties. (See "Post-deployment" on page 26.)

# Part III: Post-Deployment Configuration

This section of the guide describes the additional configuration tasks that you need to perform after LiveCycle products are deployed to the application server and the LiveCycle database is initialized (for LiveCycle Policy Server).

# 10 Configuring SSL on JBoss

This chapter describes how to create SSL credentials and configure SSL on the application server to enhance the security of communication with your application server.

**Note:** It is recommended that you complete the installation, configuration, and deployment of the LiveCycle products and ensure that the products are running correctly before configuring SSL on the application server.

It is important to ensure that the security settings configured on the application server and in the LiveCycle.ear file are consistent. If you have not already enabled SSL in the Data Manager Module (assembled as part of the LiveCycle.ear file), run Configuration Manager to reconfigure and reassemble the product with SSL enabled. The SSL password that you specify in Configuration Manager must match the password that you provide when configuring SSL on the application server. (See "Configuring LiveCycle Products" on page 44.)

**Caution:** If you are installing and deploying more than one LiveCycle product, you must consult the appropriate *Installing and Configuring* guide to obtain specific SSL and security settings for each of the LiveCycle products you have installed.

To configure SSL on JBoss, you must first create a credential using the Java keytool. You can then enable SSL on the application server by editing the jacorb.properties file in the *[appserver root]*/server/all/conf directory. Then you must edit the jboss-service file in the *[appserver root]*/server/all/conf directory.

Also note that keytool is typically located in your Java jre/bin directory. For information about using keytool, see the keytool.html file that is part of your JDK documentation.

**Note:** The password you type as the keystore password must correspond with the PassPhrase specified in the Data Manager Module. By default, this password is *bedrock*. You should change this password when you configure the Data Manager Module, but ensure that the password you enter in this step matches that password.

The value for validity, 3650, is an example. This value indicates 10 years (in days). You can set this value to the number of days appropriate to your use.

If the application server is configured to communicate with other application servers, you must add the server's certificates to the set of trusted certificates. This enables the server to confirm that it has reached the expected server when, for example, you make a policy call.

You must edit the jboss-service.xml file located in the *[appserver root]*/server/all/conf directory to ensure that data transmitted between LiveCycle Policy Server and its clients is encrypted. Additionally, this configuration allows the API to confirm the identity of the server with which it is communicating.

# Creating an SSL Credential

To configure SSL on JBoss you need an SSL credential for authentication purposes. You can use the IBM Key Management tool that is installed with Java keytool to create a credential

➤ **To create an SSL credential:**

1. From a command prompt, navigate to *[appserver root]*/server/all/svcnative, and then run the keytool using the following commands:

```
>keytool -genkey -alias ads-credentials -keyalg RSA -keystore ads-ssl.jks
-validity 3650
Enter keystore password: <password>
What is your first and last name?
[Unknown]: <first_last>
What is the name of your organizational unit?
[Unknown]: <company_name>
What is the name of your organization?
[Unknown]: <company_name>
What is the name of your City of Locality?
[Unknown]: <town_name>
What is the name of your state or province?
[Unknown]: <state_name>
What is the two-letter country code for this unit?
[Unknown]: <CA>
Is <CN=<first_last>, OU=<company_name>, O=<company_name>, L=<town_name>,
ST=<state_name>, C=<CA> correct?
[no]: yes
Enter key password for <ads-credentials>
[Unknown]: <Press ENTER if the same as keystore password>
```

**Tip:** The genkey procedure can be entered as a single command, as in the following example:

```
>keytool -genkey -alias ads-credentials -keyalg RSA -keystore ads-ssl.jks
-validity 3650 -storepass password -keypass password -dname "CN=Joe User,
OU=Joe's Group, O=Joe's Company Name, L=City Name, S=State, C=CA"
```

The new credential, ads-ssl.jks, is located in the *[appserver root]*/server/all/svcnative directory.

2. If other servers will be communicating over SSL with this server, copy the certificate you just created to ads-ssl.jks by entering the following commands:

```
>keytool -export -v -alias ads-credentials -file
"[appserver root]\server\all\svcnative\ads-ca.cer" -keystore
"[appserver root]\server\all\svcnative\ads-ssl.jks" -storepass
password
```

3. Copy the ads-ca.cer file from the *[appserver root]*/server/all/svcnative directory to the required application servers. This certificate is used in the procedure "To add certificates to the trust certificates:" on page 67.

# Enabling SSL

You can now enable SSL on the application server by editing the files indicated in this section. When SSL is configured, you must manually start JBoss.

➤ **To enable SSL:**

1. Using a text editor, open the jacorb.properties file from the *[appserver root]*/server/all/conf directory.
2. In the jacorb.properties file, replace the SSL configuration section with the following text:

```
#########################
### SSL Configuration ###
#########################
# the qualified classname of the ssl socket factory class
jacorb.ssl.socket_factory=org.jacorb.security.ssl.sun_jsse.
SSLSocketFactory
# the qualified classname of the ssl server socket factory class
jacorb.ssl.server_socket_factory=org.jacorb.security.ssl.sun_jsse.
SSLServerSocketFactory
# The name and location of the keystore. This should be absolute
# to the directory where this property file resides.
#jacorb.security.keystore=c:/jboss-3.2.5/server/all/svcnative/
ads-ssl.jks
jacorb.security.keystore=<!--replace with absolute path of ads-ssl.jks-->
jacorb.security.keystore_password=password
# trusted ca certs are also in the same keystore
jacorb.security.jsse.trustees_from_ks=on
jacorb.security.support_ssl=on
#client side ssl supported or enforced?
jacorb.security.ssl.client.supported_options=60
jacorb.security.ssl.client.required_options=0
#server side ssl supported or enforced?
jacorb.security.ssl.server.supported_options=60
jacorb.security.ssl.server.required_options=60
```

3. Enable SSL and set the keystore password in the Data Manager Module. You can configure these settings using the Configuration Manager. (See "Configuring LiveCycle Products" on page 44.)
4. Stop and restart the application server.

➤ **To add certificates to the trust certificates:**

1. Copy the cacerts file from the *[JAVA_HOME]*/jre/lib/security/ directory and store it in a safe location.
2. From a command prompt, type:

```
>keytool -import -v -noprompt -alias <uniqueAliasPerRelevantServer>
-file [appserver root]\server\all\svcnative\ads-ca.cer -keystore
"%JAVA_HOME%\jre\lib\security\cacerts"
-storepass changeit -keypass password
```

**Note:** The value `changeit` is the default password for cacerts; your actual password may be different. In a production environment, edit this password and use a strong pass phrase. You must do this as the root user.

➤ **To edit the jboss-service file:**

1. Open the jboss-service file in the *[appserver root]*/server/all/conf directory in a text editor.
2. Locate the Transaction section of the jboss-service file.
3. Add the following text:

```
<mbean code="org.jboss.security.plugins.JaasSecurityDomain"
name="jboss.security:service=JaasSecurityDomain,name=other">
   <attribute
   name="KeyStoreURL">[appserver root]/server/all/svcnative/
   ads-ssl.jks</attribute>
   <attribute name="KeyStorePass">password</attribute>
</mbean>
```

4. Replace the mbean node text below the `<!-- RMI/JRMP invoker -->` line with the following mbean node text:

```
<mbean code="org.jboss.invocation.jrmp.server.JRMPInvoker"
name="jboss:service=invoker,type=jrmp">
   <attribute name="RMIObjectPort">4444</attribute>
   <attribute name="ServerAddress">${jboss.bind.address}</attribute>
   <attribute name="RMIClientSocketFactory">org.jboss.security.ssl.
   RMISSLClientSocketFactory</attribute>
   <attribute name="RMIServerSocketFactory">org.jboss.security.ssl.
   RMISSLServerSocketFactory</attribute>
   <attribute name="SecurityDomain">java:/jaas/other</attribute>
   <depends>jboss:service=TransactionManager</depends>
   <depends>jboss.security:service=JaasSecurityDomain,name=other
   </depends>
</mbean>
```

5. Stop and restart JBoss.

# 11 Integrating with LiveCycle Policy Server

This chapter describes how to integrate LiveCycle Reader Extensions and LiveCycle Document Security with LiveCycle Policy Server. Product integration enables the following functionality:

- LiveCycle Reader Extensions can open policy-protected PDF documents using the openPDFWithAPS API method.
- LiveCycle Document Security uses the PDF Manipulation Module API to programmatically access a policy from LiveCycle Policy Server and encrypt PDF documents with the policy.
  LiveCycle Document Security can also save or decrypt a policy-encrypted document and remove the encryption from a policy-protected document.

## Configuring LiveCycle products to integrate with LiveCycle Policy Server

If LiveCycle Document Security and LiveCycle Reader Extensions are installed on a separate application server than the one where LiveCycle Policy Server is installed, you must configure the first application server to allow the two products to integrate with the LiveCycle Policy Server deployment that is running on the second application server.

If LiveCycle Document Security and LiveCycle Reader Extensions are installed on the same application server where LiveCycle Policy Server is installed, this configuration is already completed as part of the LiveCycle Policy Server configuration process.

If LiveCycle Reader Extensions or LiveCycle Document Security are installed on separate application servers, the application servers must be on the same subnet or port access might be blocked.

All products must also be running on the same application server type. For example, if LiveCycle Reader Extensions or LiveCycle Document Security is running on JBoss, then LiveCycle Policy Server must be running on JBoss.

The APS Proxy Module enables LiveCycle Document Security and LiveCycle Reader Extensions to communicate with LiveCycle Policy Server. To integrate with LiveCycle Policy Server during the configuration process, you need to indicate to LiveCycle Reader Extensions or LiveCycle Document Security the LiveCycle Policy Server host name, RMI Port, and application server initial context factory.

The values vary depending on the application server you use. You can configure these properties during the configuration process. If you are not ready to configure the integration with LiveCycle Policy Server when you install LiveCycle Reader Extensions or LiveCycle Document Security, you can use Configuration Manager to configure the APS Proxy Module properties later. Configuration Manager lets you modify the contents of the APS Proxy Module archive file without extracting the contents.

# Configuring multiple security products for integration

Perform the following configuration tasks if you are deploying a custom LiveCycle Reader Extensions or LiveCycle Document Security application that interacts with LiveCycle Policy Server hosted on JBoss:

- Create a directory called *endorsed* in the %JBOSS_HOME%/server/all/lib directory.
- Copy all of the files from the SDK endorsed directory to the %JBOSS_HOME%/server/all/lib/endorsed directory.
- Copy the jax-qname.jar file from the SDK lib directory to %JBOSS_HOME%/server/lib directory.
- Modify the JBoss startup script to include the following lines of code:

```
set JBOSS_CLASSPATH=%JBOSS_CLASSPATH%;"%JBOSS_HOME%/server/lib
/jax-qname.jar"
set JAVA_OPTS=%JAVA_OPTS% -Djava.endorsed.dirs=%JBOSS_HOME%\server
\all\ lib\endorsed
```

# Configuring LiveCycle Policy Server for EJB access

You must configure LiveCycle Document Security or LiveCycle Reader Extensions to use the LiveCycle Policy Server SDK. To perform this task, configure the LiveCycle Policy Server XML configuration file (by default named config.xml).

**Note:** When exporting the config.xml file, you must rename it to clearly identify that it is from User Management or LiveCycle Policy Server. If you overwrite the config.xml file or import the file into the wrong environment you will lose data.

This file contains the following XML tags:

```
<node name="SDK">
<map>
   <entry key="EventHandlersEnabled" value="false" />
   <entry key="WebServiceEnabled" value="false" />
   <entry key="EJBEnabled" value="false" />
</map>
</node>
```

You must set the EJBEnabled entry key to `true`. This setting enables LiveCycle Document Security or LiveCycle Reader Extensions to use the Java API. You do not need to edit the EvenHandlersEnabled and WebServiceEnabled entry keys.

For information about importing and exporting the LiveCycle Policy Server XML configuration file, see *LiveCycle Policy Server Help*.

# A Uninstalling LiveCycle Products

This section describe how to uninstall LiveCycle product files that you installed using a manual or turnkey installation.

## Removing the product files installed using a turnkey installation

Each LiveCycle product that you install using the turnkey installation includes an uninstall program that you can use to remove the product from your computer. The uninstall program removes only the specified LiveCycle product. It does not remove any files that you deployed to the application server or modules that are shared with other LiveCycle products.

Removing the product components does not remove JBoss and MySQL. You must manually remove JBoss and MySQL. When you uninstall the services, you can still start JBoss and MySQL from the command line.

**Caution:** Before removing MySQL, back up any data that you do not want to lose.

➤ **To remove the product files:**

1. Navigate to the *[product root]*/_uninst/ directory and double-click the *[product name]*_uninstall.exe file. Alternatively, you can use the Add or Remove Programs function in the Windows Control Panel to start the uninstall program.
2. If prompted, select a language for the uninstall program and click **OK**.
3. Follow the on-screen instructions, and then click **Finish**.

➤ **To remove the JBoss service:**

1. Ensure that the JBoss service is stopped.

   To stop the JBoss for Adobe LiveCycle service, from a command prompt, type:

   ```
   net stop "JBoss for Adobe LiveCycle"
   ```

2. Manually uninstall the JBoss for Adobe LiveCycle service by navigating to the *[LiveCycle root]*/jboss/ directory and, from a command prompt, typing:

   ```
   JBossService.exe -uninstall "JBoss for Adobe LiveCycle"
   ```

3. If you are not deleting the MySQL service next, delete the *[LiveCycle root]* directory.

   If you are deleting the MySQL service next, do not delete the *[LiveCycle root]* directory.

➤ **To remove the MySQL service:**

1. Ensure that the MySQL service is stopped.

   To stop the MySQL for Adobe LiveCycle service, from a command prompt, type:

   ```
   net stop "MySQL for Adobe LiveCycle"
   ```

2. Manually uninstall the MySQL for Adobe LiveCycle service by navigating to the *[LiveCycle root]*/jboss/ directory and, from a command prompt, typing:

   ```
   JBossService.exe -uninstall "MySQL for Adobe LiveCycle"
   ```

3. Delete the *[LiveCycle root]* directory.

# Removing the product files installed using manual installation

Removing the LiveCycle product installation does not remove any run-time files (that is, files that you deployed to your application server) or modules that are used by other LiveCycle products. The Uninstaller does not remove the trust directory from the installation directory.

**Caution:** In selecting to uninstall a LiveCycle product, all contents within the product install directory, including JBoss if applicable, are subject to removal without further warning. Before proceeding, back up any data you do not want to lose.

Removing the product only removes the LiveCycle *[product root]* directory from the installation directory structure.

**Note:** After you have uninstalled your product, product files will remain in the *[LiveCycle root]*/configurationManager/export directory. Do not delete these files because other LiveCycle products and the application server may be dependent on them. When you run Configuration Manager again to assemble and configure other LiveCycle products, only select the products you want to be assembled and configured.

**Tip:** On Windows, you can use the Add or Remove Programs function in the Control Panel to start the Uninstaller.

➤ **To remove the product files from your computer:**

1. Navigate to the *[LiveCycle root]*/_uninst/ directory and start the Uninstaller:
   - (Windows) Double-click the .exe file (if you are not using the Add or Remove Programs function).
   - (Linux) From a command prompt type, `file_name.bin`
2. If prompted, select a language for the uninstall program and click **OK**.
3. Follow the on-screen instructions to remove the product, and click **Finish**.
4. Delete the *[LiveCycle root]* directory.

# B Upgrading LiveCycle Products to Version 7.2 or 7.2.1

This chapter describes the tasks required to upgrade your current LiveCycle products to version 7.2, or to version 7.2.1 for LiveCycle Assembler and LiveCycle Workflow.

This document should be used in conjunction with the *Installing and Configuring LiveCycle* guide or the *Installing and Configuring LiveCycle Security Products* guide for your application server. Throughout this document, specific sections in these installing and configuring guides are listed when more detailed information is available.

For a complete list of the supported platforms and system requirements for LiveCycle 7.2 products, see the "Before You Install" chapter in this guide.

This chapter uses the following naming conventions for common file paths:

***[LiveCycle root]*** Refers to the location where LiveCycle products and components earlier than versions 7.2 or 7.2.1 are installed.

***[LiveCycle72 root]*** Refers to the location where LiveCycle products and components with versions 7.2 and 7.2.1 are installed.

The *Installing and Configuring LiveCycle* guides apply to the following products:

- LiveCycle Assembler 7.2.1
- Adobe LiveCycle Forms 7.2
- Adobe LiveCycle Form Manager 7.2
- Adobe LiveCycle PDF Generator 7.2
- Adobe LiveCycle Print 7.2
- Adobe LiveCycle Workflow 7.2.1
- Adobe LiveCycle Workflow Designer 7.2.1
- Watched Folder 1.1

The *Installing and Configuring LiveCycle Security Products* guides apply to the following products:

- Adobe LiveCycle Document Security 7.2
- Adobe LiveCycle Reader Extensions 7.2
- Adobe LiveCycle Policy Server 7.2

## Upgrade guidelines

This section describes guidelines you must follow when upgrading LiveCycle products to version 7.2 or version 7.2.1.

### Updating your application server

Ensure that you apply the required patches and fix packs to the application server on which you are running LiveCycle products, and obtain the updated database drivers. Your environment must meet the system requirements described in the "Support software" chapter in this guide.

### Using automatic or turnkey installations for upgrading

For installations for deployment to WebSphere or WebLogic, you can use Configuration Manager to configure and deploy the product, initialize the database, and verify the deployment.

Before using Configuration Manager to automatically configure LiveCycle products, the existing LiveCycle components must be undeployed and the application server restarted.

For installations for deployment to JBoss, you can configure the products for deployment and initialize the database using Configuration Manager, but you must deploy the product components manually. The turnkey installation option is only supported for upgrading LiveCycle Reader Extensions.

If you are running LiveCycle products that were originally installed using the turnkey installation option, you can upgrade to version 7.2 or 7.2.1 by following the manual or the auto-configuration upgrade instructions that are included in this guide. Instructions for upgrading LiveCycle Reader Extensions using the turnkey installation option are also included in this chapter.

### Configuring using Configuration Manager during the upgrade process

It is recommended that you do not automatically configure the WebSphere or WebLogic application server using Configuration Manager because current configuration settings on your application server may be overwritten.

During the upgrade process, you will configure some product and application server run-time properties using Configuration Manager. For upgrades from 7.x versions, you should configure the upgraded product using the same property values. The "Configuring LiveCycle Products for Deployment" chapter in the *Installing and Configuring* guides provides information about the properties that you will configure when you upgrade the product.

## Summary of manual upgrade process

This checklist describes the high-level tasks that you must perform to upgrade from a LiveCycle 7.x product to a LiveCycle 7.2 or 7.2.1 product. For detailed information, see the specific upgrade procedures for the product you are upgrading.

The sections referenced in the "See" column are in the *Installing and Configuring* guides for your application server, depending on the product you are installing.

| Task | See |
|---|---|
| Back up the database that contains the current LiveCycle configuration and run-time data. | The database server documentation. |
| Back up copies of the currently deployed LiveCycle EAR and WAR files to a separate directory. | |

| Task | See |
|---|---|
| (JBoss) Stop services, if applicable (JBoss for Adobe LiveCycle, MySQL for Adobe LiveCycle, JBoss for Workflow BAM, AdobeDocumentSecurity, or AdobeReaderExtensions). | |
| Undeploy LiveCycle components from the application server. | Refer to the application server documentation. |
| Uninstall the previous version of LiveCycle products using the uninstaller program. | The "Uninstalling LiveCycle Products" chapter |
| Apply the required patches and fix packs to the application server and obtain the updated database drivers. | The "Supported software" section of the "Before You Install" chapter |
| Install the LiveCycle 7.2 or 7.2.1 product to a new (non-default) location on your file system. Do not install to the same directory where your previous LiveCycle products are installed. | The "Installing LiveCycle Products" chapter |
| Run Configuration Manager to configure the product. Apply the configuration data that you used in the original installation. For WebSphere and WebLogic, you can automatically deploy the product, initialize the database, and verify the deployment using Configuration Manager. | (JBoss) The "Configuring LiveCycle Products for Deployment" chapter<br>(WebLogic, WebSphere) The "Automatically Configuring LiveCycle Products" or "Configuring LiveCycle Products for Deployment" chapter |
| (JBoss) Deploy the product components to the application server. | The "Manually deploying to JBoss" chapter |
| (JBoss) Run Configuration Manager to initialize the database. | The "Initializing the Database" chapter |

# LiveCycle Forms and LiveCycle Print

This section provides upgrading instructions for LiveCycle Forms 7.2 and LiveCycle Print 7.2.

**Note:** You must install LiveCycle Forms 7.2 before installing LiveCycle Print 7.2.

## Upgrading from Adobe Form Server 6.0 to LiveCycle Forms 7.2

This procedure describes how to upgrade from Adobe Form Server 6.0 to LiveCycle Forms 7.2.

➤ **To upgrade from Form Server 6.0 to LiveCycle Forms 7.2:**

1. Undeploy the earlier product from the application server. (For information, see your application server documentation.)

2. Uninstall the earlier product using the uninstaller program. (For information, see the Form Server 6.0 *Installation and Configuration Guide*.)

3. Upgrade the application server and database to ensure that they meet the system and software requirements for LiveCycle 7.2 products.

4. Install and deploy LiveCycle Forms 7.2, and initialize the database using the instructions in this document for the manual installation and deployment option. For WebSphere and WebLogic, you can automatically deploy the product, initialize the database, and verify the deployment using Configuration Manager.

**Note:** If you did not include User Management with the installation and configuration of LiveCycle Forms, you do not need to connect to a database, and you do not need to follow steps to initialize the database.

# LiveCycle Forms and LiveCycle Print 7.x to LiveCycle Forms and LiveCycle Print 7.2

This procedure describes how to upgrade from LiveCycle Forms 7.0 or 7.1 to LiveCycle Forms 7.2, and from LiveCycle Print 7.1 to LiveCycle Print 7.2.

It is recommended that you install LiveCycle Forms 7.2 to a new directory so that you do not overwrite the previously installed version.

➤ **To retrieve form configuration properties (for LiveCycle Forms 7.0 or 7.1 configured without User Management):**

1. Type the following URL in a web browser:

   `http://[host_name]:[port]/FormServerAdmin/settings.html`

   The default port number for WebLogic is 7001. If you are running a Managed Server, you may have configured the application server to use a different port number, such as 8001. The default port for WebSphere is 9080; the default port for JBoss is 8080.

2. Record the settings that appear on this page.

➤ **To retrieve form configuration properties (for LiveCycle Forms 7.0 or 7.1 configured *with* User Management):**

1. Type the following URL in a web browser:

   `http://[host_name]:[port]/adminui`

   The default port number for WebLogic is 7001. If you are running a Managed Server, you may have configured the application server to use a different port number, such as 8001. The default port for WebSphere is 9080; the default port for JBoss is 8080.

2. Log into Administrator.

3. Click **Services**, and then click **Adobe LiveCycle Forms**.

4. Record the settings that appear on this page.

5. These properties must be set for the new installation on the Form Server Module Configuration screen on Configuration Manager when you install and configure LiveCycle Forms 7.2.

➤ **To upgrade to LiveCycle Forms 7.2 and LiveCycle Print 7.2:**

1. Back up the database that currently contains the LiveCycle Forms 7.0 or 7.1 configuration and run-time data.

2. Ensure that you have a back-up copy of the currently deployed LiveCycle EAR and WAR files that are configured for the current production system that you are planning to upgrade.

3. Back up the XDC files from LiveCycle Print 7.1.

4. (JBoss) If your LiveCycle deployment is running on a JBoss Application Server with a MySQL database that you installed using the turnkey option, stop the JBoss for Adobe LiveCycle service and the MySQL for Adobe LiveCycle service.

5. Undeploy the following files from your application server:
   - LiveCycle.ear
   - FormsIVS.ear
   - adobe-FontManager.war
   - LCMBootstrapper.war
   - adobe-printSubmitter.ear (LiveCycle Print only)

   For information about undeploying from WebLogic or WebSphere, see "Uninstalling EAR files" in the *Installing and Configuring LiveCycle* guide.

   **Note:** When the Livecycle.ear file is undeployed, LiveCycle Print 7.1 does not work.

6. Stop the WebLogic Server or WebSphere Application Server.

7. Uninstall the previous versions of LiveCycle products using the uninstaller program. (See "Uninstalling LiveCycle Products" in the related *Installing and Configuring LiveCycle* guide.)

8. Upgrade the application server and database to ensure that they meet the system and software requirements for LiveCycle Forms 7.2.

9. Install LiveCycle Forms 7.2 to a new (non-default) directory (for example, C:\Adobe\LiveCycle72\ **or** /opt/adobe/livecycle72/). Follow the instructions in "Installing LiveCycle Products" in the *Installing and Configuring LiveCycle* guide.

   **Note:** If you are installing or upgrading multiple LiveCycle 7.2 products, be sure to install them to the same root *[LiveCycle72 root]* directory.

10. Copy the DocumentServicesLibrary.jar file, according to your application server:
    - (WebLogic) From the *[LiveCycle72 root]*/components/csa/weblogic/lib/adobe directory to the *[appserverdomain]*/lib directory.
    - (WebSphere) From the *[LiveCycle72 root]*/components/csa/websphere/lib/adobe directory to the *[appserver root]*/optionalLibraries.
    - (JBoss) From the *[LiveCycle72 root]*/components/csa/jboss/lib/adobe directory to the *[appserver root]*/server/all/lib directory.

    **Note:** Modify the XDC files installed with LiveCycle Print 7.2 to match those that you are using with version 7.1, and use these modified files. For information about the XDC files included with LiveCycle Print, see the *Getting Started* guide and the *Editing XDC Files to Customize Printing Workflows* guide for LiveCycle Print.

11. Start the application server.

12. Configure LiveCycle Forms 7.2 using Configuration Manager. (See the "Configuring LiveCycle Products for Deployment" chapter in the *Installing and Configuring LiveCycle* guide.) As you proceed through the Configuration Manager screens, choose the following options:
    - **Configuration Mode**: Select **Custom Configuration Wizard**.
    - **Product Selection**: Select the application server you are using, as well as **Foundation**, **LiveCycle Forms**, and **LiveCycle Print** (if applicable).
    - **Adobe User Management Selection**: Select one of the following options:
      - **LiveCycle Forms with User Management and Administrator** if you used LiveCycle Forms with User Management in the previous deployment
      - **LiveCycle Forms without User Management and Administrator** if you did not previously use LiveCycle Forms with User Management
    - **Task Selection**: Select the following options:
      - **Configure and Assemble products**
      - **Bootstrap Database** (only if you are including User Management in the configuration)

      For WebSphere and WebLogic, also select these options:
      - **Deploy products**
      - **Verify deployed products**

13. Follow the instructions on the remaining Configuration Manager screens. Accept the default values in Configuration Manager whether you are configuring with User Management and Administrator or without User Management and Administrator.

    **Note:** When prompted to specify the Global Storage Directory location, specify the same location that you currently use for this directory.

    If you are deploying to WebSphere or WebLogic, complete steps 14 and 15. If you are deploying to JBoss, complete steps 16 and 17.

14. (WebSphere and WebLogic) On the Confirm Products to Deploy screen, select these EAR files:
    - LiveCycle.ear
    - adobe-FontManager.ear.
    - adobe-printSubmitter.ear (LiveCycle Print only)

15. (WebSphere and WebLogic) Follow the instructions on the Configuration Manager screens to initialize the database and verify the deployed products.

16. (JBoss) Deploy LiveCycle Forms 7.2. (See "Manually Deploying to JBoss" in the *Installing and Configuring LiveCycle* guide.)

17. (JBoss) (User Management configuration) Run Configuration Manager to initialize the database. Select **Custom Configuration Wizard**, and then select **Bootstrap** database.

    **Note:** Initializing the database is necessary to add new table columns to the database schema. Initializing the database does not alter existing data.

18. Deploy the FormsIVS.ear files according to the steps required for your application server. (See "LiveCycle Forms post-deployment tasks" in the *Installing and Configuring LiveCycle* guide.)

19. Verify the deployment by following the instructions in "LiveCycle Forms post-deployment tasks" in the *Installing and Configuring LiveCycle* guide.

20. Update your application's class path with the location of these JAR files: formserver-client.jar, adobe-common.jar, datamanager-client.jar, and AdobeCSAUtils.jar. Add the um-client.jar file to the application class path if the application passes an `InvocationContext` object in the Form Server Module API. For details about these files, see Developing Custom Applications located in the *[LiveCycle72 root]*/forms/documentation directory.

➤ **To configure LiveCycle Forms (when not using User Management):**

1. Type the following URL in a web browser:

   `http://[host_name]:[port]/FormServerAdmin/settings.html`

   The default port number for WebLogic is 7001. If you are running a Managed Server, you may have configured the application server to use a different port number, such as 8001. The default port for WebSphere is 9080; the default port for JBoss is 8080.

2. Enter the settings that you recorded when you retrieved the LiveCycle Forms configuration settings from the previous LiveCycle.ear file. (See "To retrieve form configuration properties (for LiveCycle Forms 7.0 or 7.1 configured without User Management):" on page 76.)

3. Click **Save**.

   **Note:** The settings modified here are not retained when the application server is restarted.

➤ **To configure LiveCycle Forms (when using User Management):**

1. Type the following URL in a web browser:

   `http://[host_name]:[port]/adminui`

2. The default port number for WebLogic is 7001. If you are running a Managed Server, you may have configured the application server to use a different port number, such as 8001. The default port for WebSphere is 9080; the default port for JBoss is 8080.

3. Log into Administrator.

4. Click **Services**, and then click **Adobe LiveCycle Forms**.

5. Enter the settings you recorded when you retrieved the LiveCycle Forms configuration settings from the previous LiveCycle.ear file. (See "To retrieve form configuration properties (for LiveCycle Forms 7.0 or 7.1 configured with User Management):" on page 76.)

6. Click **Save**.

# LiveCycle Form Manager

This section provides instructions for upgrading from LiveCycle Form Manager 7.0.1 to LiveCycle Form Manager 7.2. To perform this upgrade, you must update the server components that are deployed to the application server and initialize the database.

It is recommended that you install LiveCycle Form Manager 7.2 to a new directory so that you do not overwrite the previously installed version.

➤ **To upgrade LiveCycle Form Manager 7.0.1 to 7.2:**

1. Back up the database that currently contains the LiveCycle Form Manager 7.0.1 configuration and run-time data.
2. Ensure that you have a back-up copy of the currently deployed LiveCycle EAR and WAR files that are configured for the current production system that you are planning to upgrade.
3. (JBoss) If your LiveCycle deployment is running on a JBoss Application Server with a MySQL database that you installed using the turnkey option, stop the JBoss for Adobe LiveCycle service and the MySQL for Adobe LiveCycle service.
4. Undeploy the following LiveCycle Form Manager 7.0.1 components by following the instructions in the *Installing and Configuring* guide for LiveCycle Form Manager 7.0.1:
   - adobe-FontManager.war
   - LiveCycle.ear
   - LCMBootstrapper.war

   For information about undeploying from WebLogic or WebSphere, see "Uninstalling EAR files" in this *Installing and Configuring LiveCycle* guide.
5. Stop the WebLogic Server or WebSphere Application Server.
6. Uninstall the previous version of LiveCycle products using the uninstaller program. (See "Uninstalling LiveCycle Products" in the related *Installing and Configuring LiveCycle* guide.)
7. Upgrade the application server and database to ensure that they meet the system and software requirements for LiveCycle 7.2 products.
8. Install LiveCycle Form Manager 7.2 to a new (non-default) directory (for example, C:\Adobe\LiveCycle72\ **or** /opt/adobe/livecycle72/). Follow the instructions in "Installing LiveCycle Products" in the *Installing and Configuring LiveCycle* guide.

   **Note:** If you are installing or upgrading multiple LiveCycle 7.2 products, be sure to install them to the same *[LiveCycle72 root]* directory.
9. Copy the DocumentServicesLibrary.jar file, according to your application server:
   - (WebLogic) From the *[LiveCycle72 root]*/components/csa/weblogic/lib/adobe directory to the *[appserverdomain root]*/lib directory
   - (WebSphere) From the *[LiveCycle72 root]*/components/csa/websphere/lib/adobe directory to the *[appserver root]*/optionalLibraries directory
   - (JBoss) From the *[LiveCycle72 root]*/components/csa/jboss/lib/adobe directory to the *[appserver root]*/server/all/lib directory
10. Start the application server.
11. Configure LiveCycle Form Manager 7.2 using Configuration Manager. (See "Configuring LiveCycle Products for Deployment" in the *Installing and Configuring LiveCycle* guide.) As you proceed through the Configuration Manager screens, choose the following options:
    - **Configuration Mode**: Select **Custom Configuration Wizard**.
    - **Product Selection**: Select the application server you are using, as well as **Foundation** and the product you are configuring.

- **Task Selection**: Select the following options:
  - **Configure and Assemble products**
  - **Bootstrap Database**

  For WebSphere and WebLogic, also select these options:
  - **Deploy products**
  - **Verify deployed products**

12. Follow the instructions on the Configuration Manager screens to configure the LiveCycle.ear file. Ensure that you configure your LiveCycle 7.2 EAR file with the same settings that you chose for version 7.0.1. If you are deploying to WebSphere or WebLogic, complete steps 13 and 14. If you are deploying to JBoss, complete steps 15 and 16.

    **Note:** When prompted to specify the Global Storage Directory location, specify the same location that you currently use for this directory.

13. (WebSphere and WebLogic) On the Confirm Products to Deploy screen, select these EAR files:
    - LiveCycle.ear
    - adobe-FontManager.ear

14. (WebSphere and WebLogic) Follow the instructions on the Configuration Manager screens to initialize the database and verify the deployed products.

15. (JBoss) Deploy LiveCycle Form Manager 7.2 components. (See "Manually Deploying to JBoss" in the *Installing and Configuring LiveCycle* guide.)

16. (JBoss) Run Configuration Manager to initialize the database. Select **Custom Configuration Wizard**, and then select **Bootstrap database**. (See "Initializing the Database" in the *Installing and Configuring LiveCycle* guide.)

    **Note:** Initializing the database is necessary to add new table columns to the database schema. Initializing the database does not alter existing data.

17. Verify the deployment by following the instructions in "LiveCycle Form Manager post-deployment tasks" the *Installing and Configuring LiveCycle* guide.

    **Note:** You can also verify your installation and configuration by using the samples available at www.adobe.com/devnet/livecycle/samples.html.

# LiveCycle Assembler, LiveCycle Workflow, and Watched Folder

This section describes how to upgrade any or all of the following products:

- LiveCycle Workflow 7.0.1 or 7.2 to LiveCycle Workflow 7.2.1
- LiveCycle Assembler 7.2 to LiveCycle Assembler 7.2.1
- Watched Folder 1.0 to Watched Folder 1.1

If you are upgrading LiveCycle Workflow as a stand-alone product, do not include the steps that pertain to Watched Folder or LiveCycle Assembler. Watched Folder is intended for use with LiveCycle Assembler and LiveCycle Workflow; you must install LiveCycle Workflow before installing Watched Folder.

To perform this upgrade, you must update the server components that are deployed to the application server and initialize the database. It is recommended that you install LiveCycle Assembler 7.2.1 to a new directory so that you do not overwrite the previously installed version.

When you configure LiveCycle Assembler 7.2.1 using Configuration Manager, ensure that you set the Security Groups option to the same value that you have set for LiveCycle Assembler 7.2. (The value can be set to an asterisk (*), which only allows users with a valid entry in the LDAP system to log in, or to no value, which allows any user to log in.)

➤ **To upgrade to LiveCycle Assembler 7.2.1, LiveCycle Workflow 7.2.1, and Watched Folder 1.1:**

1. Back up the database that currently contains the LiveCycle Workflow 7.0.1 or 7.2 configuration and run-time data.

2. Ensure that you have a back-up copy of the currently deployed LiveCycle EAR and WAR files that are configured for the current production system that you are planning to upgrade.

3. (JBoss) If your LiveCycle deployment is running on a JBoss Application Server with a MySQL database that you installed using the turnkey option, stop the JBoss for Adobe LiveCycle service and the MySQL for Adobe LiveCycle service.

4. Undeploy the following files from your application server:
   - LiveCycle.ear
   - adobe-FontManager.war
   - LCMBootstrapper.war
   - adobe-Assembler7.ear (LiveCycle Assembler only)

   For information about undeploying from WebLogic or WebSphere, see "Uninstalling EAR files" in the *Installing and Configuring LiveCycle* guide.

5. (WebSphere and WebLogic) Stop the application server.

6. Uninstall the previous versions of LiveCycle products using the uninstaller program. (See "Uninstalling LiveCycle Products" in the *Installing and Configuring LiveCycle* guide.)

7. Install LiveCycle Assembler 7.2.1 to a new (non-default) directory (for example, C:\Adobe\LiveCycle72\ **or** /opt/adobe/livecycle72/). Follow the instructions in "Installing LiveCycle Products" in the *Installing and Configuring LiveCycle* guide.

8. Install LiveCycle Workflow 7.2.1 to the same directory where you installed LiveCycle Assembler 7.2.1 by following the instructions in "Installing LiveCycle Products" in the related *Installing and Configuring LiveCycle* guide.

   **Note:** You can install LiveCycle Assembler and LiveCycle Workflow in any order.

9. Install Watched Folder to the same directory where you installed LiveCycle Workflow by following the instructions in "Installing Watched Folder" in the *Installing and Configuring LiveCycle* guide. LiveCycle Workflow must already be installed.

   **Note:** If you are installing or upgrading multiple LiveCycle 7.2 products, be sure to install them to the same *[LiveCycle72 root]* directory.

10. Copy the DocumentServicesLibrary.jar file, according to your application server:
    - (WebLogic) From the *[LiveCycle72 root]*/components/csa/weblogic/lib/adobe directory to the *[appserverdomain]*/lib directory.
    - (WebSphere) From the *[LiveCycle72 root]*/components/csa/websphere/lib/adobe directory to the *[appserver root]*/optionalLibraries.
    - (JBoss) From the *[LiveCycle72 root]*/components/csa/jboss /lib/adobe directory to the *[appserver root]*/server/all/lib directory. Copy the adobe-service.xml file from the *[LiveCycle72_root]*/configurationManager/deploy/jboss directory to the *[appserver root]*/server/all/deploy/jms directory.

11. Start the application server.

12. Configure the LiveCycle products by using Configuration Manager. (See "Configuring LiveCycle Products for Deployment" in the *Installing and Configuring LiveCycle* guide.) As you proceed through the Configuration Manager screens, choose the following options:
    - **Configuration Mode**: Select **Custom Configuration Wizard**.
    - **Product Selection**: Select the application server you are using, and then select **Foundation**, as well as all of the products that you installed.
    - **Task Selection**: Select the following options:
        - **Configure and Assemble products**
        - **Bootstrap Database**

      For WebSphere and WebLogic, also select these options:
        - **Deploy products**
        - **Verify deployed products**

13. Follow the instructions on the remaining Configuration Manager screens. Ensure that you configure your 7.2.1 EAR file with the same settings you chose for version 7.0.1 or 7.2. If you are deploying to WebSphere or WebLogic, complete steps 14 and 15. If you are deploying to JBoss, complete steps 16 and 17.

    **Note:** When prompted to specify the Global Storage Directory location, specify the same location that you currently use for this directory.

14. (WebSphere and WebLogic) On the **Confirm Products to Deploy** screen, select the archives that you are deploying:
    - adobe-FontManager.ear
    - LiveCycle.ear
    - adobe-Assembler7.ear (LiveCycle Assembler only)

15. (WebSphere and WebLogic) Follow the instructions on the Configuration Manager screens to initialize the database and verify the deployed products.

16. (JBoss) Deploy LiveCycle Assembler, LiveCycle Workflow, and Watch Folder 7.2. (See "Manually Deploying to JBoss" in the *Installing and Configuring LiveCycle* guide.) These components consist of these EAR files:
    - LiveCycle.ear
    - adobe-FontManager.ear
    - LCM.ear
    - adobe-Assembler7.ear

17. (JBoss) Run Configuration Manager to reinitialize the database. Select **Custom Configuration Wizard**, and then select **Bootstrap database**. (See "Initializing the Database" in the *Installing and Configuring LiveCycle* guide.)

18. Upgrade the Assembler QPAC. (See *Upgrading LiveCycle Workflow Designer* or the *Creating Workflows* guide, available with the LiveCycle Workflow documentation set.)

    **Note:** Initializing the database is necessary to add new table columns to the database schema. Initializing the database does not alter existing data.

19. Verify the deployment by following the instructions in the "Post Deployment" section in the *Installing and Configuring LiveCycle* guide.

    **Note:** You can also verify your installation and configuration by using the samples available at www.adobe.com/devnet/livecycle/samples.html.

# LiveCycle Workflow Designer

You need to use the version of LiveCycle Workflow Designer that LiveCycle Workflow 7.2.1 provides. If you want to use the new User QPAC with existing workflows, you need to migrate your workflows.

You must upgrade your LiveCycle QPACs from LiveCycle Workflow 7.0.1 or 7.2 to version 7.2.1. See "Updating components" in the *Creating Workflows* guide that is included with LiveCycle Workflow Designer.

➤ **To upgrade LiveCycle Workflow Designer:**

1. Uninstall LiveCycle Workflow 7.0.1 Designer.

2. Install LiveCycle Workflow 7.2.1 Designer. (See "Installing LiveCycle Workflow Designer" in the *Installing and Configuring LiveCycle* guide.)

   **Note:** LiveCycle Workflow 7.2.1 provides an updated User QPAC that includes new features. If any of your workflows include User actions and you want to use the new features, you must migrate your workflows. For more information, see the topic "Migrating Workflows from Older Versions" in the *Creating Workflows* guide or in *LiveCycle Workflow Designer Help*.

# BAM Server for LiveCycle Workflow

This section describes general best practices to follow when upgrading to a new version of BAM Server. You must use the BAM Server upgrade utility to upgrade metadata XML files that you export from LiveCycle Workflow Business Activity Monitor during the upgrade process.

Upgrading BAM Server involves performing the following tasks:

- Exporting the BAM metadata using BAM Workbench
- Upgrading the BAM metadata using the BAM Server upgrade utility
- Undeploying the BAM Server EAR file from the application server
- Removing recovery log files and dropping existing tables into the BAM metadata database
- Deploying the BAM Server EAR file that LiveCycle Workflow 7.2.1 provides
- Importing the upgraded BAM metadata using BAM Workbench

## About the BAM Server upgrade utility

The BAM Server upgrade utility upgrades exported metadata files so that they include new features required for the current product version.

The BAM Server upgrade utility is packaged in the cqupgrade.jar file. When you install LiveCycle Workflow, this file is located in the *[LiveCycle root]*/Workflow/bam/CQUpgrade directory.

**Note:** The upgrade utility generates the message "Error parsing input file" when files cannot be upgraded. If this error is generated, the file specified in the error message may be damaged. Contact Adobe Systems for assistance.

### Syntax

The following command upgrades metadata files that have been exported to a JAR file:

```
java -jar cqupgrade.jar -jar -i inJar.jar [-o outJar.jar]
```

### Parameters

You can use the following parameters in the BAM Server upgrade utility command:

***inJar.jar***: The path to the JAR file that contains the exported metadata to upgrade.

***outJar.jar*** (optional): The path to the upgraded JAR file. If you omit this parameter, the upgraded files are saved in the JAR file specified in *inJar.jar*.

The following examples show commands that you can use to upgrade metadata stored in JAR files using the BAM Server upgrade utility:

- The following command upgrades the metadata files that are stored in a JAR file named toUpgrade.jar, which is located in the same directory as cqupgrade.jar:

  ```
  java -jar cqupgrade.jar -i toUpgrade.jar
  ```

- The following command upgrades the metadata files in the toUpgrade.jar file and saves the upgraded files to a different JAR file named upgraded.jar. The file is saved in the same directory as cqupgrade.jar:

  ```
  java -jar cqupgrade.jar -i toUpgrade.jar -o upgraded.jar
  ```

➤ **To upgrade Business Activity Monitor:**

1. Log into BAM Workbench.
2. Click the **Administration** tab and then click **Import/Export**.
3. In the **Operations** menu, select **Export Metadata to a JAR File (download)**.
4. In the File Download dialog box, click **Save**.
5. Specify a location and file name for the exported JAR file, and click **Save**.
6. Click **Close** in the Download Complete dialog box.
7. Click **System Settings**, and then click the **Checkpoint Configuration** tab. Note the directory path in the **Recovery Log Directory** box. You will need to know where the recovery log directory is at a later step in this procedure.
8. Shut down Business Activity Monitor.

9. Use the BAM Server upgrade utility (cqupgrade.jar) to upgrade the metadata that you exported in a JAR file. (See "About the BAM Server upgrade utility" on page 85.)

10. Undeploy the BAM Server EAR file by following the instructions for your application server:
    - (JBoss) Remove the following directories:
      - *[jboss bam root]*/server/default/work
      - *[jboss bam root]*/server/default/tmp
      - *[jboss bam root]*/server/default/data
      - *[jboss bam root]*/server/conf/jboss.web
    - (WebSphere) Undeploy the old BAM Server EAR file using WebSphere Administrative Console.
    - (WebLogic) Undeploy the old BAM Server EAR file using WebLogic Server Administration Console.

11. Delete all of the files from the recovery log directory that have names similar to the following patterns:
    - filestore*.dat
    - DEFAULTRECOVERYLOGGER_*
    - chkpoint_.x

12. Use your database management tools to drop the database tables that store the BAM metadata. Alternatively, you may want to create a new BAM metadata database. For more information about the BAM metadata database, see "Creating the BAM metadata database" in the *Installing and Configuring LiveCycle* guide for your application server.

13. Deploy the new version of BAM Server to the application server. (See "Deploying BAM Server" in the *Installing and Configuring LiveCycle* guide.)

14. Log into BAM Workbench.

15. Review the BAM Server configuration settings to determine if updates are required. (See "Getting Started with BAM Server" in the *Installing and Configuring LiveCycle* guide.)

16. Click the **Administration** tab and then click **Import/Export**.

17. In the **Operations** menu, select **Import Metadata from a JAR File (upload)**.

18. Click **Browse** to locate the JAR file that contains the upgraded BAM metadata, and then click **OK**.

19. Restart the BAM Server instance.

# LiveCycle PDF Generator

To upgrade from LiveCycle PDF Generator 7.0.1 or 7.0.2 to LiveCycle PDF Generator 7.2, you must undeploy the product you are currently using before installing and deploying the new product.

The instructions in this section apply to LiveCycle PDF Generator for PostScript, LiveCycle PDF Generator Elements, and LiveCycle PDF Generator Professional.

➤ **To upgrade LiveCycle PDF Generator 7.0.2 to LiveCycle PDF Generator 7.2:**

1. Back up the database that currently contains the LiveCycle PDF Generator 7.0.1 or 7.0.2 configuration and run-time data.

2. Ensure that you have a back-up copy of the currently deployed LiveCycle EAR and WAR files that are configured for the current production system that you are planning to upgrade.

3. (JBoss) If your LiveCycle PDF Generator deployment is running on a JBoss Application Server with a MySQL database that you installed using the turnkey option, stop the JBoss for Adobe LiveCycle service.

4. If you are installing LiveCycle PDF Generator Professional or LiveCycle PDF Generator Elements, uninstall Adobe Acrobat 7.0.5 from the Add/Remove programs window in the Microsoft Windows Control Panel, and then reboot your system.

5. Undeploy the following LiveCycle PDF Generator components by following the instructions in the *Installing and Configuring* guide for LiveCycle PDF Generator 7.0.2:

   - pdfg-all.ear (or pdfg-ps-all.ear)
   - LiveCycle.ear
   - adobe-FontManager.war
   - LCMBootstrapper.war

   For information about undeploying from WebLogic or WebSphere, see "Uninstalling EAR files" in the *Installing and Configuring LiveCycle* guide.

6. (WebSphere and WebLogic) Stop the application server.

7. Uninstall the previous versions of LiveCycle products by using the uninstaller program. (See "Uninstalling LiveCycle Products" in the *Installing and Configuring LiveCycle* guide.)

8. Install LiveCycle PDF Generator 7.2 to a new (non-default) directory (for example, C:\Adobe\LiveCycle72\ **or** /opt/adobe/livecycle72/. Follow the instructions in "Installing LiveCycle Products" in the *Installing and Configuring LiveCycle* guide.

   **Note:** If you are installing or upgrading multiple LiveCycle 7.2 products, be sure to install them to the same *[LiveCycle72 root]* directory.

9. Copy the DocumentServicesLibrary.jar file, according to your application server:

   - (WebLogic) From the *[LiveCycle72 root]*/components/csa/weblogic/lib/adobe directory to the *[appserverdomain]*/lib directory
   - (WebSphere) From the *[LiveCycle72 root]*/components/csa/websphere/lib/adobe directory to the *[appserver root]*/optionalLibraries
   - (JBoss) From the *[LiveCycle72 root]*/components/csa/jboss/lib/adobe directory to the *[appserver root]*/server/all/lib directory. Copy the adobe-service.xml file from the *[LiveCycle72_root]*/configurationManager/deploy/jboss directory to the *[appserver root]*/server/all/deploy/jms directory.

10. Start the application server.

11. Configure the LiveCycle products by using Configuration Manager. (See "Configuring LiveCycle Products for Deployment" in the *Installing and Configuring LiveCycle* guide.) As you proceed through the Configuration Manager screens, choose the following options:
    - **Configuration Mode**: Select **Custom Configuration Wizard**.
    - **Product Selection**: Select the application server you are using, and then select **Foundation**, and **LiveCycle PDF Generator**.
    - **Task Selection**: Select all of the following options:
        - **Configure and Assemble products**
        - **Bootstrap Database**

      For WebSphere and WebLogic, also select these options:
        - **Deploy products**
        - **Verify deployed products**

12. Follow the instructions on the remaining Configuration Manager screens. Ensure that you configure your LiveCycle 7.2 EAR file with the same settings you chose for version 7.0.2. If you are deploying to WebSphere or WebLogic, complete steps 13 and 14. If you are deploying to JBoss, complete steps 15 and 16.

    **Note:** When prompted to specify the Global Storage Directory location, specify the same location that you currently use for this directory.

13. (WebSphere and WebLogic) On the Confirm Products to Deploy screen, select the archives that you are deploying:
    - pdfg-all.ear (or pdfg-ps-all.ear)
    - adobe-FontManager.ear
    - LiveCycle.ear

14. (WebSphere and WebLogic) Follow the instructions on the Configuration Manager screens to initialize the database and verify the deployed products.

15. (JBoss) Deploy the LiveCycle PDG Generator 7.2 components:
    - LiveCycle.ear
    - adobe-FontManager.ear
    - LCM.ear
    - pdfg-all.ear (or pdfg-ps-all.ear)

    (See "Manually Deploying to JBoss" in the *Installing and Configuring LiveCycle* guide.)

16. (JBoss) Run Configuration Manager to reinitialize the database. Select **Custom Configuration Wizard**, and then select **Bootstrap database**. (See "Initializing the Database" in the *Installing and Configuring LiveCycle* guide.)

    **Note:** Initializing the database is necessary to add new table columns to the database schema. Initializing the database does not alter existing data.

17. Verify the deployment by following the instructions in "LiveCycle PDF Generator post-deployment tasks" in the *Installing and Configuring LiveCycle* guide.

    **Note:** You can also verify your installation and configuration by using the samples available at www.adobe.com/devnet/livecycle/samples.html.

➤ **To configure JMS Messaging for WebSphere running on Windows:**

1. In the WebSphere Administrative Console, delete the JMS queues, topics, and listener ports for each server.
2. Configure WebSphere MQ by following the instructions in "Configuring JMS resources for WebSphere MQ" in the *Installing and Configuring LiveCycle* guide.
3. Run the scripts provided by Configuration Manager to create queues in the MQ installation by following the instructions in "Preparing WebSphere MQ" in the *Installing and Configuring LiveCycle* guide.
4. Run Configuration Manager again and select the **Configure the Application Server** task. Follow the instructions on the Configuration Manager screens to configure application server details. (See "Changing the application server settings" in the *Installing and Configuring LiveCycle* guide.) As you proceed through the Configuration Manager screens, choose the following options:
   - **JVM Settings**: Select **Do Not Apply**.
   - **JMS settings**: Specify values for the properties of the MQ service.
   - **Datasource Creation**: Select **Do Not Apply**.
5. On the Configure Application Server screen, click **Apply Settings Now** to configure your application server, and, when the task is complete, click **Next**.
6. Restart the application server and, on the Application Server/Cluster Restart screen, click **Next** if you selected another task for Configuration Manager to perform, or click **Finish** to close Configuration Manager.
7. Verify that the applications start correctly.

# LiveCycle Document Security

This section provides instructions for upgrading from LiveCycle Document Security 7.0 or 7.1 to LiveCycle Document Security 7.2. To perform this upgrade, you must update the server components that are deployed to the application server.

It is recommended that you install LiveCycle Document Security 7.2 to a new directory so that you do not overwrite the previously installed version.

➤ **To upgrade to LiveCycle Document Security 7.2:**

1. Back up the following files and folders:
   - ../trust.xml
   - ../credentials/
   - ../certificates/
   - ../CRLs/
   - ../keystore
   - ../trust.sig

2. Ensure that you have a back-up copy of the currently deployed LiveCycle EAR and WAR files that are configured for the current production system that you are planning to upgrade.

3. Undeploy the following LiveCycle Document Security components by following the instructions in the *Installing and Configuring* guide for LiveCycle Document Security 7.0 or 7.1:

| **WebSphere and WebLogic** | **JBoss** |
|---|---|
| adobe-FontManager.war | adobe-FontManager.bar |
| adobe-PDFManipulation.war | adobe-PDFManipulation.bar |
| adobe-TrustManager.war | adobe-TrustManager.bar |
| DataManagerService.war | AdobeServices.sar |
| ServicesNatives-2.war | ServicesNatives-2.war |
| adobe-APSProxy.war | adobe-APSProxy.bar |

4. (WebSphere) Set up default users, roles, and login files by following in the instructions in the procedure "To map users to roles" in the *Installing and Configuring LiveCycle Security Products* guide. (This procedure must be completed each time the LiveCycle-security.ear file is redeployed.)

5. Restart the application server.

6. Uninstall the previous version of LiveCycle products by using the uninstaller program. (See "Uninstalling LiveCycle Products" in the related *Installing and Configuring LiveCycle* guide.)

7. Install LiveCycle Document Security 7.2 to a new (non-default) directory (for example, C:\Adobe\LiveCycle72\ **or** /opt/adobe/livecycle72/. Follow the instructions in "Installing LiveCycle Products" in the *Installing and Configuring LiveCycle Security Products* guide.

   **Note:** If you are installing or upgrading multiple LiveCycle 7.2 products, be sure to install them to the same *[LiveCycle72 root]* directory.

8. Copy the DocumentServicesLibrary.jar file, according to your application server:
   - (JBoss) From the *[LiveCycle72 root]*/components/csa/jboss/lib/adobe directory to the *[appserver root]*/server/all/lib directory.
   - (WebLogic) From the *[LiveCycle72 root]*/components/csa/weblogic/lib/adobe directory to the *[appserver domain]*/lib directory.
   - (WebSphere) From the *[LiveCycle72 root]*/components/csa/websphere/lib/adobe directory to the *[appserver root]*/optionalLibraries directory.

9. Configure LiveCycle Document Security 7.2 using Configuration Manager. (See "Configuring LiveCycle Products".) As you proceed through the Configuration Manager screens, choose the following options:
   - **Configuration Mode**: Select **Custom Configuration Wizard**.
   - **Product Selection**: Select the application server you are using, as well as **Foundation** and the product you are configuring.
   - **Task Selection**: Select the following options:
     - **Configure and Assemble products**
     - **Bootstrap Database**

     For WebSphere and WebLogic, also select these options:
     - **Deploy products**
     - **Verify deployed products**
   - **Trust Directory Selection**: Select **Create a new trust directory**.
   - **Keystore Selection**: Select **Create a new keystore and pair**.
10. Follow the instructions on the remaining Configuration Manager screens. Ensure that you configure your 7.2 EAR file with the same settings you chose for version 7.0 or 7.1. If you are deploying to WebSphere or WebLogic, complete steps 11 and 12. If you are deploying to JBoss, complete steps 13 and 14.

    **Note:** When prompted to specify the Global Storage Directory location, specify the same location that you currently use for the "Directory for Adobe LiveCycle products" temp file.
11. (WebSphere and WebLogic) On the Confirm Products to Deploy screen, select the archives that you are deploying:
    - adobe-FontManager.ear
    - LiveCycle.ear
    - LiveCycle-security.ear
12. (WebSphere and WebLogic) Follow the instructions on the Configuration Manager screens to initialize the database and verify the deployed products.
13. (JBoss) Deploy LiveCycle Document Security 7.2. (See "Manually Deploying to JBoss" in the *Installing and Configuring LiveCycle Security Products* guide.)
14. (JBoss) Run Configuration Manager to reinitialize the database. Select **Custom Configuration Wizard**, and then select **Bootstrap database**. (See "Initializing the Database" in the *Installing and Configuring LiveCycle Security Products* guide.)
15. Restart the application server.
16. Verify your installation and configuration by using the samples available at www.adobe.com/devnet/livecycle/samples.html.

# LiveCycle Reader Extensions

This section provides instructions for upgrading from LiveCycle Reader Extensions 7.0 or 7.0.2 to LiveCycle Reader Extensions 7.2. To perform this upgrade, you must update the server components that are deployed to the application server.

**Note:** (JBoss) If you previously installed LiveCycle Reader Extensions on Windows for deployment to JBoss using the turnkey method, see "To upgrade LiveCycle Reader Extensions using the turnkey method" in the *Installing and Configuring LiveCycle Security Products* guide for the turnkey upgrade instructions.

It is recommended that you install LiveCycle Reader Extensions 7.2 to a new directory so that you do not overwrite the previously installed version.

If you are using an existing credential, ensure the credential has not expired and is still valid. (See "LiveCycle Reader Extensions Rights credential" in the *Installing and Configuring LiveCycle Security Products* guide.)

Custom applications using existing APIs supported in Adobe Document Services 6.0 for Reader Extensions are not supported in LiveCycle Reader Extensions 7.2. Custom applications using existing APIs supported in Adobe Reader Extensions Server 6.1 are supported by and continue to work with LiveCycle Reader Extensions 7.2.

Digital certificates issued for use with Reader Extensions Server 6.1 are compatible for use with LiveCycle Reader Extensions 7.2. If you are upgrading from version 6.1 to 7.2, you will not receive a new Rights credential.

➤ **To upgrade LiveCycle Reader Extensions to 7.2:**

1. Back up the following files and folders in the *[LiveCycle root]*/ReaderExtensions/trust directory:
   - ../trust.xml
   - ../credentials/
   - ../certificates/
   - ../CRLs/
   - ../keystore
   - ../trust.sig
2. Ensure that you have a back-up copy of the currently deployed LiveCycle EAR and WAR files that are configured for the current production system that you are planning to upgrade.
3. Undeploy the following LiveCycle Reader Extensions components by following the instructions in the *Installing and Configuring* guide for LiveCycle Reader Extensions 7.0 or 7.0.2:

| WebSphere and WebLogic | JBoss |
|---|---|
| adobe-FontManager.war | adobe-FontManager.bar |
| adobe-PDFManipulation.war | adobe-PDFManipulation.bar |
| adobe-TrustManager.war | adobe-TrustManager.bar |
| DataManagerService.war | AdobeServices.sar |
| ServicesNatives-2.war | ServicesNatives-2.war |

| WebSphere and WebLogic | JBoss |
|---|---|
| ares.ear | ares.ear |
| adobe-CredentialSecurityEJB.ear | adobe-CredentialSecurityEJB.ear |
| adobe-APSProxy.war | adobe-APSProxy.bar |

4. Restart the application server.
5. Uninstall the previous version of LiveCycle products y using the uninstaller program. (See "Uninstalling LiveCycle Products" in the related *Installing and Configuring LiveCycle* guide.)
6. Install LiveCycle Reader Extensions 7.2 to a new (non-default) directory (for example, C:\Adobe\LiveCycle72\ **or** /opt/adobe/livecycle72/. Follow the instructions in "Installing LiveCycle Products" in the *Installing and Configuring LiveCycle Security Products* guide.

   **Note:** If you are installing or upgrading multiple LiveCycle 7.2 products, be sure to install them to the same *[LiveCycle72 root]* directory.
7. Copy the DocumentServicesLibrary.jar file, according to you application server:
   - (WebLogic) From the *[LiveCycle72 root]*/components/csa/weblogic/lib/adobe directory to the *[appserver domain]*/lib directory.
   - (WebSphere) From the *[LiveCycle72 root]*/components/csa/websphere/lib/adobe directory to the *[appserver root]*/optionalLibraries directory.
   - (JBoss) From the *[LiveCycle72 root]*/components/csa/jboss/lib/adobe directory to the *[appserver root]*/server/all/lib directory.
8. Configure LiveCycle Reader Extensions 7.2 by using Configuration Manager. (See "Configuring LiveCycle Products" in the *Installing and Configuring LiveCycle Security Products* guide.) As you proceed through the Configuration Manager screens, choose the following options:
   - **Configuration Mode**: Select **Custom Configuration Wizard**.
   - **Product Selection**: Select the application server you are using, as well as **Foundation** and the product you are configuring.
   - **Task Selection**: Select the following options:
     - **Configure and Assemble products**

     For WebSphere and WebLogic, also select these options:
     - **Deploy products**
     - **Verify deployed products**
   - **Trust Directory Selection**: Select **Create a new trust directory**.
   - **Keystore Selection**: Select **Create a new keystore and pair**.
9. Follow the instructions on the remaining Configuration Manager screens. Ensure that you configure your 7.2 EAR file with the same settings you chose for version 7.0 or 7.1. If you are deploying to WebSphere or WebLogic, complete steps 10 and 11. If you are deploying to JBoss, complete step 12.

   **Note:** When prompted to specify the Global Storage Directory location, specify the same location that you currently use for the "Directory for Adobe LiveCycle products" temp file.

10. (WebSphere and WebLogic) On the Confirm Products to Deploy screen, select the archives that you are deploying:
    - adobe-FontManager.ear
    - LiveCycle.ear
    - LiveCycle-security.ear
11. (WebSphere and WebLogic) Follow the instructions on the Configuration Manager screens to initialize the database and verify the deployed products.
12. (JBoss) Deploy LiveCycle Reader Extensions 7.2. (See "Manually Deploying to JBoss" in the *Installing and Configuring LiveCycle Security Products* guide.)
13. Restart the application server.
14. Verify the installation and configuration by going to the appropriate URL:
    - (WebLogic) http://*[host_name]*:7001/ReaderExtensions **or** http://*[host_name]*:8001/ReaderExtensions (Managed Server)
    - (WebSphere) http://*[host_name]*:9080/ReaderExtensions
    - (JBoss) http://*[host_name]*:8080/ReaderExtensions
15. Type the user name and password you created when you configured the LiveCycle Reader Extensions user. (See "Setting up users, roles, and login files" for JBoss or WebLogic or "To map users to roles" for WebSphere in the *Installing and Configuring LiveCycle Security Products* guide.)

➤ **(JBoss) To upgrade LiveCycle Reader Extensions using the turnkey method:**

1. Stop the AdobeReaderExtensions Windows service from the Services window in the Administrative Tools area of the Windows Control Panel.
2. Verify that port 8080 is not being used.
3. The turnkey installation specifies "localhost" as the host and "8080" as the port for use by JBoss. If JBoss is already installed, ensure that it is not using port 8080. You cannot configure an alternative host or port for JBoss during the turnkey installation and configuration process.
4. Install LiveCycle Reader Extensions 7.2 using the turnkey method to a new (non-default) directory (for example, C:\Adobe\LiveCycle72\). Follow the instructions in "Installing LiveCycle Reader Extensions or LiveCycle Document Security" in the *Installing and Configuring LiveCycle Security Products* guide.
5. Verify the installation and configuration by going to the URL http://*[host_name]*:8080/ReaderExtensions.

   **Note:** For turnkey installations, the default user name is administrator and the default password is password. To edit these values, see "Setting up users, roles, and login files" in the *Installing and Configuring LiveCycle Security Products* guide.

   **Note:** You can also verify your installation and configuration by using the samples available at www.adobe.com/devnet/livecycle/samples.html.

# LiveCycle Policy Server

This section provides instructions for upgrading from LiveCycle Policy Server 7.0.2 to LiveCycle Policy Server 7.2. To perform this upgrade, you must update the server components that are deployed to the application server and initialize the database.

It is recommended that you install LiveCycle Policy Server 7.2 to a new directory so that you do not overwrite the previously installed version.

➤ **To upgrade LiveCycle Policy Server to 7.2:**

1. Back up the database that currently contains the LiveCycle Policy Server 7.0.2 configuration and run-time data.
2. Ensure that you have a back-up copy of the currently deployed LiveCycle EAR and WAR files that are configured for the current production system that you are planning to upgrade.
3. Undeploy the following LiveCycle Policy Server 7.0.2 components:
   - asn1.jar
   - jsafe.jar
   - jsafeJCE.jar
   - edc-server-spi.jar
   - edc-server.ear
4. Remove the dom*.jar files from the *[appserver root]*/java/jre/lib/endorsed directory.
5. Uninstall the previous version of LiveCycle products by using the uninstaller program. (See "Uninstalling LiveCycle Products" in the relevant *Installing and Configuring LiveCycle* guide.)
6. Install LiveCycle Policy Server 7.2 to a new (non-default) directory (for example, C:\Adobe\LiveCycle72\ **or** /opt/Adobe/livecycle72/). Follow the instructions in "Installing LiveCycle Products" in the *Installing and Configuring LiveCycle Security Products* guide.

   **Note:** If you are installing or upgrading multiple LiveCycle 7.2 products, be sure to install them to the same *[LiveCycle72 root]* directory.
7. Deploy LiveCycle Policy Server 7.2. (See the chapter about manually deploying to the application server in the *Installing and Configuring LiveCycle Security Products* guide for your application server.)
8. Run Configuration Manager to reinitialize the database. Select **Custom Configuration Wizard**, and then select **Bootstrap database**. (See "Initializing the Database" in the *Installing and Configuring LiveCycle Security Products* guide.)

   **Note:** Initializing the database is necessary to add new table columns to the database schema. Initializing the database does not alter existing data.
9. Configure various run-time settings for LiveCycle Policy Server. (See the "Post-deployment" section of the *Installing and Configuring LiveCycle Security Products* guide.)

   **Note:** You can verify your installation and configuration by using the samples available at www.adobe.com/devnet/livecycle/samples.html.

# C Enhancing Server Performance

This appendix contains general tips that you can use to improve server performance when using LiveCycle products.

## Optimizing inline documents and impacts on JVM memory

If you are typically processing documents of a relatively small size, you can improve the performance associated with the document transfer speed and storage space by implementing the following LiveCycle product configurations:

- Increase the maximum inline size for LiveCycle products so that it is larger than the size of most documents.
- For processing larger files, specify storage directories that are located on a high-speed disk system or a RAM disk.

The default maximum inline size and the storage directories (the Adobe LiveCycle products temporary file directory and the global storage directory) are properties of the Data Manager Module. You can configure the Data Manager Module using Configuration Manager. (See "Configuring LiveCycle Products" on page 44.)

**Note:** The default maximum inline size value is 65536 bytes.

### Document size and maximum inline size

When a document that is sent for processing by LiveCycle products is less than or equal to the maximum inline size, the document is stored on the server inline and the document is serialized as an Adobe Document object. Storing documents inline can have significant performance benefits.

A document that is larger than the maximum inline size is stored on the local file system (in the storage directories specified using Configuration Manager), and the Adobe Document object that is transferred to and from the server is only a pointer to that file.

### JVM maximum heap size

An increase in the maximum inline size requires more memory for storing the serialized documents, and therefore generally also requires an increase in the JVM maximum heap size. The maximum JVM heap size should not exceed 2GB.

A heavily-loaded system that is processing a large number of documents can rapidly saturate the JVM heap memory. To avoid an OutOfMemoryError, the JVM maximum heap size must be increased by an amount corresponding to the size of the inline documents multiplied by the number of documents that are typically executed at any given time.

To calculate how much the JVM maximum heap must be increased, using the following equation:

JVM maximum heap size increase= (inline documents size) x (average number of documents processed)

**Example C.1** ***Calculating the JVM maximum heap size***

In this example, the current JVM maximum heap is set to 512 MB, and the maximum inline size is 64 KB. The server needs to be configured for the scenario where ten jobs are run simultaneously, and each job has nine input files and one result file (a total of ten files per job, and 1000 files processed simultaneously). All of the files are under 512 KB in size.

To store all of the files inline, the maximum inline size must be set to at least 512 KB.

The required increase in the JVM maximum heap size is calculated using the following equation:

(512 KB) x (100) = 51200 KB, or 51.2 MB

The JVM maximum heap size must be increased by 512 MB for a total of 1GB.

### Considering heap fragmentation

Setting the size of inline documents to large values raises the risk of OutOfMemoryError on systems that are prone to heap fragmentation. To store a document inline, there must be sufficient contiguous space in the JVM heap memory. Some operating systems, JVMs, and garbage collection algorithms are prone to heap fragmentation. Fragmentation decreases the amount of contiguous heap space, and can lead to OutOfMemoryError even when sufficient total free space exists.

For example, previous operations on the application server have left the JVM heap in a fragmented state, and the garbage collector is unable to compact the heap sufficiently to regain large blocks of free space. OutOfMemoryError can occur even though the JVM maximum heap size has been adjusted for an increase in maximum inline size.

To account for heap fragmentation, the inline document size must not be set higher than 0.1% of the total heap size. For example, a JVM maximum heap size of 512 MB can support a maximum inline size of 512 MB x 0.001 = 0.512 MB, or 512 KB.

## Cleaning up temporary files from Global storage directory

If the Global storage directory is not set explicitly in Configuration Manager, the default location of the Global storage directory is *[TempDir]*/AdobeDocumentStorage/global. If *[TempDir]* is also not specified by the user in Configuration Manager, the default location is java.io.tmpdir.

# D Supported Platform and Software Combinations

This appendix provides the supported platforms and software combinations for LiveCycle products. For a summary of the platforms combinations, see "Before You Install" on page 9.

| Platform | Application server | JDK | Database |
|---|---|---|---|
| Microsoft Windows Server 2003, Standard Edition<br>Microsoft Windows Server 2003, Enterprise Edition | JBoss 3.2.5 | J2SDK version 1.4.2_04 | IBM® DB2 8.2 (Version 8.1 FixPack 7) |
| | JBoss 3.2.5 | J2SDK version 1.4.2_04 | MS SQL Server 2000 SP3 |
| | JBoss 3.2.5 | J2SDK version 1.4.2_04 | MySQL 4.1 |
| | JBoss 3.2.5 | J2SDK version 1.4.2_04 | Oracle 9i |
| | JBoss 3.2.5 | J2SDK version 1.4.2_04 | Oracle 10g |
| | IBM WebSphere® 5.1.1.5 | IBM JDK installed with WebSphere | Oracle 9i |
| | BEA WebLogic Server® 8.1 SP5 | J2SDK version 1.4.2_08 | Oracle 9i |
| | BEA WebLogic Server 8.1 SP5 | J2SDK version 1.4.2_08 | Oracle 10g |
| | IBM WebSphere 5.1.1.5 | IBM JDK installed with WebSphere | Oracle 10g |
| | BEA WebLogic Server 8.1 SP5 | J2SDK version 1.4.2_08 | IBM DB2 8.2 (Version 8.1 FixPack 7) Applicable to LiveCycle Policy Server only. |
| | IBM WebSphere 5.1.1.5 | IBM JDK installed with WebSphere | IBM DB2 8.2 (Version 8.1 FixPack 7) |
| | IBM WebSphere 5.1.1.5 | IBM JDK installed with WebSphere | MS SQL Server 2000 SP3 |

| Platform | Application server | JDK | Database |
|---|---|---|---|
| Red Hat Linux Advanced Server 2.1 Update 3 | JBoss 3.2.5 | J2SDK version 1.4.2_04 | IBM DB2 8.2 (Version 8.1 FixPack 7) |
| | JBoss 3.2.5 | J2SDK version 1.4.2_04 | MS SQL Server 2000 SP3 |
| | JBoss 3.2.5 | J2SDK version 1.4.2_04 | MySQL 4.1 |
| | JBoss 3.2.5 | J2SDK version 1.4.2_04 | Oracle 9i |
| | JBoss 3.2.5 | J2SDK version 1.4.2_04 | Oracle 10g |
| | BEA WebLogic Server 8.1 SP5 | J2SDK version 1.4.2_08 | IBM DB2 8.2 (Version 8.1 FixPack 7) Applicable to LiveCycle Policy Server only. |
| | BEA WebLogic Server 8.1 SP5 | J2SDK version 1.4.2_08 | Oracle 9i |
| | BEA WebLogic Server 8.1 SP5 | J2SDK version 1.4.2_08 | Oracle 10g |
| | IBM WebSphere 5.1.1.5 | IBM JDK installed with WebSphere | IBM DB2 8.2 (Version 8.1 FixPack 7) |
| | IBM WebSphere 5.1.1.5 | IBM JDK installed with WebSphere | Oracle 9i |
| | IBM WebSphere 5.1.1.5 | IBM JDK installed with WebSphere | Oracle 10g |

| Platform | Application server | JDK | Database |
|---|---|---|---|
| Red Hat Linux Advanced Server 3.0 | JBoss 3.2.5 | J2SDK version 1.4.2_04 | IBM DB2 8.2 (Version 8.1 FixPack 7) |
| | JBoss 3.2.5 | J2SDK version 1.4.2_04 | MySQL 4.1 |
| | JBoss 3.2.5 | J2SDK version 1.4.2_04 | Oracle 9i |
| | JBoss 3.2.5 | J2SDK version 1.4.2_04 | Oracle 10g |
| | BEA WebLogic Server 8.1 SP5 | J2SDK version 1.4.2_08 | IBM DB2 8.2 (Version 8.1 FixPack 7) Applicable to LiveCycle Policy Server only. |
| | BEA WebLogic Server 8.1 SP5 | J2SDK version 1.4.2_08 | Oracle 9i |
| | BEA WebLogic Server 8.1 SP5 | J2SDK version 1.4.2_08 | Oracle 10g |
| | IBM WebSphere 5.1.1.5 | IBM JDK installed with WebSphere | IBM DB2 8.2 (Version 8.1 FixPack 7) |
| | IBM WebSphere 5.1.1.5 | IBM JDK installed with WebSphere | Oracle 9i |
| | IBM WebSphere 5.1.1.5 | IBM JDK installed with WebSphere | Oracle 10g |
| IBM AIX® 5.2 | IBM WebSphere 5.1.1.5 | IBM JDK installed with WebSphere | Oracle 9i |
| | IBM WebSphere 5.1.1.5 | IBM JDK installed with WebSphere | Oracle 10g |
| IBM AIX 5.3 | IBM WebSphere 5.1.1.5 | IBM JDK installed with WebSphere | IBM DB2 8.2 (Version 8.1 FixPack 7) |
| | IBM WebSphere 5.1.1.5 | IBM JDK installed with WebSphere | Oracle 9i |
| | IBM WebSphere 5.1.1.5 | IBM JDK installed with WebSphere | Oracle 10g |

| Platform | Application server | JDK | Database |
|---|---|---|---|
| Sun™ Solaris™ 8 | BEA WebLogic Server 8.1 SP5 | J2SDK version 1.4.2_08 | IBM DB2 8.2 (Version 8.1 FixPack 7) Applicable to LiveCycle Policy Server only. |
| | BEA WebLogic Server 8.1 SP5 | J2SDK version 1.4.2_08 | Oracle 9i |
| | BEA WebLogic Server 8.1 SP5 | J2SDK version 1.4.2_08 | Oracle 10g |
| | IBM WebSphere 5.1.1.5 | IBM JDK installed with WebSphere | IBM DB2 8.2 (Version 8.1 FixPack 7) |
| | IBM WebSphere 5.1.1.5 | IBM JDK installed with WebSphere | Oracle 9i |
| | IBM WebSphere 5.1.1.5 | IBM JDK installed with WebSphere | Oracle 10g |
| Sun Solaris 9 | BEA WebLogic Server 8.1 SP5 | J2SDK version 1.4.2_08 | IBM DB2 8.2 (Version 8.1 FixPack 7) Applicable to LiveCycle Policy Server only. |
| | BEA WebLogic Server 8.1 SP5 | J2SDK version 1.4.2_08 | Oracle 9i |
| | IBM WebSphere 5.1.1.5 | IBM JDK installed with WebSphere | Oracle 9i |
| | IBM WebSphere 5.1.1.5 | IBM JDK installed with WebSphere | Oracle 10g |

| Platform | Application server | JDK | Database |
|---|---|---|---|
| SUSE Linux Enterprise Server 9.0 | JBoss 3.2.5 | J2SDK version 1.4.2_04 | IBM DB2 8.2 (Version 8.1 FixPack 7) |
| | JBoss 3.2.5 | J2SDK version 1.4.2_04 | MS SQL Server 2000 SP3 |
| | JBoss 3.2.5 | J2SDK version 1.4.2_04 | Oracle 9i |
| | JBoss 3.2.5 | J2SDK version 1.4.2_04 | Oracle 10g |
| | JBoss 3.2.5 | J2SDK version 1.4.2_04 | MySQL 4.1 |
| | BEA WebLogic Server 8.1 SP5 | J2SDK version 1.4.2_08 | IBM DB2 8.2 (Version 8.1 FixPack 7) Applicable to LiveCycle Policy Server only. |
| | BEA WebLogic Server 8.1 SP5 | J2SDK version 1.4.2_08 | Oracle 9i |
| | IBM WebSphere 5.1.1.5 | IBM JDK installed with WebSphere | IBM DB2 8.2 (Version 8.1 FixPack 7) |
| | IBM WebSphere 5.1.1.5 | IBM JDK installed with WebSphere | Oracle 9i |
| | IBM WebSphere 5.1.1.5 | IBM JDK installed with WebSphere | Oracle 10g |

# E Content and Format of the trust.xml File

***This chapter applies to LiveCycle Reader Extensions and LiveCycle Document Security only.***

The trust.xml file contains most of the trust information needed by the PDF Manipulation Module. Any trust information not directly contained in the file is referred to by the file.

The trust.xml file consists of the following sections:

- trustAnchors (certificates)
- credentials
- CRLs
- prefs (preferences)

These sections can be arranged in any order, and each section consists of record elements pertaining to that section. There can be multiple instances of each section. Preference names and values are all case-sensitive.

This chapter describes the content and format of each section of the trust. xml file.

## Example of the trust.xml file

The following text is an example of a typical trust. xml file:

```
<?xml version="1.0" encoding="UTF-8" ?>
<trust>
   <trustAnchors>
      <cerrecord cerFile="JohnSmithDER.cer" TrustedFor="Signatures
      CertifiedDocuments DynamicContent Identity SSL" />
      <cerrecord cerFile="Alice.cer" TrustedFor="Identity Signatures" />
   </trustAnchors>
   <CRLs>
  <crl URL="http://crl.adobe.com/testCA3.crl" filename="testCA3.crl" />
  <crl URL="http://crl.adobe.com/testCA4.crl" filename="testCA4.crl" />
  <crl URL="http://crl.adobe.com/cds.crl" filename="cds.crl" />
   </CRLs>
   <credentials>
      <p12record alias="alice" p12="alice.pfx" EmbedRevInfo="false"
      TimestampURL="http://uname:password@tsa.com/tsa" />
      <p12record alias="bob" p12="bob.pfx" />
      <p12record alias="charlie" p12="Charlie.pfx" />
      <p12record alias="doug" p12="Doug.pfx" />
      <hsmrecord alias="Alicehsm" slot="17" dllpath="/dev/null" />
      <p12record alias="GoodUbiquity" p12="gretsky-alpha-2.pfx" />
      <p12record alias="johnsmith" p12="johnsmith.pfx" />
      <MSCAPIrecord alias="AliceCertify"
      sha1="c4500e618f4fc1cef417020bb3638bd82a78ccec" />
      <p12record alias="UBFormsEval" p12="UBFormsEval.pfx" />
   </credentials>
```

```
  <prefs>
    <proxy name="network-proxy" port="80" />
    <ocsp URL="" SendNonce="true" ReqRevCheck="CertRevRequiredIfInfoAvail"
    MaxClockSkew="10" ResponseFreshness="525600"
    URLToConsult="AIAInCertToCheck" SignRequest="false" />
    <timestamp URL="http://tsp.adobe.com" CheckRevocation="CertRev"
    HashAlgorithm="SHA1" Username="uname" Password="pwd" />
    <signature EmbedRevInfo="true" VerificationTime="UseSigningTime" />
  </prefs>
</trust>
```

# trustAnchor element

The `trustAnchor` element has one sub-element, which is `cerrecord`. The `cerrecord` sub-element has two required attributes:

- `cerFile`—A reference to a certificate (.cer) file. The value for this attribute must be a file name only, not a path name.
- `trustedFor`—A list of things that the certificate is trusted for.

This information is used during the signature validation operation of the PDF Manipulation Module, and is used in a similar manner as the AddressBook in Acrobat. The PDF Manipulation Module searches the certificates directory for file names. The certificates directory is imported into the deployment unit. The Adobe root CDS certificate, from which all Adobe CDS intermediate CAs are issued, is always automatically trusted for the `Identity`, `Signatures`, and `CertifiedDocuments` flags.

The values allowed in the `trustedFor` attribute are described in this table.

| Flag | Description |
|---|---|
| `Identity` | Include this certificate when determining trust. If this flag is not present, the certificate can be used in building a certificate chain, but cannot be used to determine what the signature is trusted for. |
| `Signatures` | Documents signed with this signature, or whose certificate chain includes this certificate, are trusted. |
| `CertifiedDocuments` | Documents signed with this signature as an author signature, or whose certificate chain includes this certificate, are considered trusted for `CertifiedDocuments`. |
| `DynamicContent` | This value is valid only when the `CertifiedDocuments` flag is also present in the flag list. When present, dynamic content (movies, audio, and so on) are allowed. |
| `EmbeddedJavaScript` | This value is valid only when the `CertifiedDocuments` flag is also present in the flag list. When present, JavaScript™ embedded in the document can be executed. This flag needs to be set when you want to allow certified documents to run any embedded Javascript scripts on the server. |
| `SSL` | (Linux) Use this certificate as a trust anchor when determining if any particular server is trusted for SSL communication. In Windows, the default (Internet Explorer) certificate store is used instead. |

# credentials element

The `credentials` element has three sub-elements that describe a type of private key container:

- `p12record`—A reference to a PKCS#12 (.p12) file stored on disk.
- `hsmrecord`—A reference to an HSM.
- `MSCAPIrecord`—A reference to an entry in the Microsoft database (on systems that run Windows).

Passwords required to access the private keys are supplied through the API and are not included in the trust.xml file. The PDF Manipulation Module searches the credentials directory for file names. The credentials directory is imported into the deployment unit.

The available record types and corresponding attributes are described in this table.

| Record type | Attributes | Description |
|---|---|---|
| `p12record` | `alias` | The name by which the credential is known to the PDF Manipulation Module API. It must be unique in the credentials section of the trust.xml file. |
| | `p12file` | The PKCS#12 file name. It is searched for among the files imported into the deployment unit. |
| | `sha1` | (Optional) The SHA1 fingerprint of the corresponding certificate. The `sha1` value can be used to distinguish among different keys if more than one is stored in a single PKCS#12 file. If the `sha1` value is not provided and the PKCS#12 file contains multiple appropriate credentials, an exception is raised. |
| `hsmrecord` | `alias` | The name by which the credential is known to the PDF Manipulation Module API. It must be unique in the credentials section of trust.xml. |
| | `dllpath` | The location of the DLL in the file system. For HSM support, a DLL is required that implements the PKCS#11 interface for that particular HSM. |
| | `slot` | The slot number that identifies where the private key is stored on the HSM. |
| | `sha1` | (Optional) The SHA1 fingerprint of the corresponding certificate. The `sha1` value can be used to distinguish among different keys if more than one is stored in a single PKCS#12 file. If the sha1 value is not provided and the PKCS#12 file contains multiple appropriate credentials, an exception is raised. |
| `MSCAPIrecord` | `alias` | The name by which the credential is known to the PDF Manipulation Module API. It must be unique in the credentials section of the trust.xml file. |
| | `sha1` | The SHA1 fingerprint of the corresponding certificate. This value must be used to select among the different credentials stored in the Microsoft certificate store. |

| Record type | Attributes | Description |
|---|---|---|
| Common to all record types | `EmbedRevInfo` | (Optional) Specifies whether the revocation information is embedded within the certificate. The value `true` indicates that, if available, the revocation information is embedded. The default value is `false`.<br>The value of this attribute overrides the global preference. (See the `signature` record type under the `prefs` element in the table that begins on page 107.) |
| | `TimestampURL` | (Optional) The URL to consult for timestamping information for this credential. The URL must contain the user name and password, if required, in this format: http(s)://*[username]*:*[password]*@*[path]*<br>The value of this attribute overrides the global preference. (See the `signature` record type under the `prefs` element in the table that begins on page 107.) |

# CRL element (Certificate revocation)

The `CRL` element lists all of the CRL files used by the PDF Manipulation Module for certificate revocation checking. The `CRL` element uses the record type `CRL`. The attributes of the records map from a URL (referenced by the `CRLdp` value in a certificate) to a file name where the actual CRL is stored so that LiveCycle Document Security or LiveCycle Reader Extensions never directly fetches a CRL from the web. The system administrator must keep the CRL files updated. The file referenced by the `filename` attribute is searched for in the directory specified using Configuration Manager.

The attributes of a `CRL` record are described in this table.

| Attribute | Description |
|---|---|
| `URL` | A reference to the `CRLdp` value in the corresponding certificate. It must exactly match the URL found in the `CRLdp` field of the certificate. |
| `filename` | The file name of the CRL. |

During the initialization of the PDF Manipulation Module, if a CRL file is missing for a URL entry in the trust.xml file, a warning is generated in the log file. The warning alerts the administrator to install any missing CRL files.

If no matching URL is found for a `CRLdp` value in the trust.xml file during signature validation, LiveCycle Document Security or LiveCycle Reader Extensions looks up the CRL over the network. If the CRL is not retrieved, the signature validation fails and a warning is not generated in the log file.

# prefs element (Plug-in preferences)

The `prefs` element provides global preferences for all of the operations performed by the PDF Manipulation Module. Some of the global preferences may be overridden by information within certificates or credentials, or by individual certificate or credential preferences (as described in the previous sections). The `prefs` element has four sub-elements:

- `timestamp`
- `ocsp`
- `signature`
- `proxy`

The attributes of each sub-element are described in the following table.

| Record type | Attribute | Description |
|---|---|---|
| `timestamp` | `URL` | (Optional) The timestamp server to be used for timestamping digital signatures. It can contain a user name and password, if required, in this format:<br>http(s)://*[username]*:*[password]*@[*path]*<br>The value of this attribute is overridden by the `credential` value if present for a credential. |
| | `Username` | This attribute is required if a timestamp server is specified. If this value is present in addition to a username value in the `URL` attribute, the value in the `URL` attribute is tried first. If that value fails, the value specified by this `Username` attribute is used. |
| | `Password` | This attribute is required if a timestamp server is specified. If this value is present in addition to a password value in the `URL` attribute, the value in the `URL` attribute is tried first. If that value fails, the value specified by this `Password` attribute is used. |
| | `CheckRevocation` | (Optional) Specifies whether revocation checking on the timestamp server certificate is on or off. You can set this attribute to one of the following values:<br>• `Never`: Never checks.<br>• `BestEffort`: Try to if possible. No error if no revocation information is available.<br>• `RequiredIfInfoAvail`: Revocation information is returned if available.<br>• `AlwaysRequired`: Revocation checking is always required.<br>The default value is `RequiredIfInfoAvail`. |
| | `HashAlgorithm` | (Optional) This attribute is used for request and response verification. You can set this attribute to one of the following values: `MD5`, `SHA1`, or `SHA256`. The default value is `SHA1`. |

| Record type | Attribute | Description |
|---|---|---|
| ocsp | URL | (Optional) The local OCSP Server URL. This attribute is used only when the URLToConsult attribute (see attribute later in this table) is set to LocalConfig or LocalConfigIfNoAIA. |
| | SendNonce | (Optional) Specifies whether to send a random number in the OCSP request to prevent replay attacks. The default value is true. |
| | CheckRevocation | (Optional) Specifies whether the revocation checking on the OCSP certificates is turned on or off. You can set this attribute to one of the following values:<br>• Never: Never checks.<br>• BestEffort: Try to if possible. No error if no revocation information is available.<br>• RequiredIfInfoAvail: Revocation information is returned if available.<br>• AlwaysRequired: Revocation checking is always required.<br>The default value is RequiredIfInfoAvail. |
| | MaxClockSkew | (Optional) The maximum allowed skew in response time and local time (in minutes). The default value is 5. |
| | ResponseFreshness | (Optional) The maximum time validity of a preconstructed OCSP response (in minutes). The default value is 525600 (one year). |
| | URLToConsult | (Optional) The URL to be used for OCSP checking. You can set the attribute to one of the following values:<br>• AIAInCertToCheck: Use URL from the certificate.<br>• LocalConfig: Use the local URL provided using the URL pref. (See the previous description of OCSP URL, in this table.)<br>• LocalConfigIfNoAIA: Use local URL if none is provided in the certificate.<br>The default value is AIAInCertToCheck. The default value indicates that the URL should be present in the certificate. |
| | SignRequest | (Optional) Specifies whether to sign the request. The default value is false. |

| Record type | Attribute | Description |
|---|---|---|
| `signature` | `EmbedRevInfo` | (Optional) Specifies whether to embed revocation information within the signature. A value of `true` indicates the revocation information is embedded, if available. The default value is false.<br>The value of this attribute is overridden by the credential value if present for a credential. (See the "Common to all record types" section of the table that begins on page 105.) |
| | `VerificationTime` | (Optional) The time to use for signature verification. You can set this attribute to one of the following values:<br>• `AlwaysUseCurrentTime`: Use current time.<br>• `UseSigningTimeIfSecureElseCurrent`: Use signing time if it is secure (timestamped); otherwise, use current time.<br>• `UseSigningTime`: Use signing time.<br>The default value is `UseSigningTimeIfSecureElseCurrent`. |
| `proxy` | `Host` | The host name or IP address of the proxy server. |
| | `Port` | (Optional) The port where the proxy server is running. The default value is `80`. |
| | `Username` | (Optional) The user name to be provided when authenticating to the proxy server. If the user name is not provided, authentication to the proxy server is not performed. |
| | `Password` | (Optional) The password to be provided when authenticating to the proxy server. If the password is not provided, authentication to the proxy server is not performed. |

# F Fonts Installed with the Font Manager Module

Some modules require access to fonts that are installed with the Font Manager Module. This module contains a number of bundled fonts that you can use in your custom applications.

For information on adding your own fonts to the Font Manager Module, see "Configuring LiveCycle Products" on page 44.

The following fonts are installed with the Font Manager Module:

- Letter Gothic Std Medium
- Letter Gothic Std Slanted
- Letter Gothic Std Bold
- Letter Gothic Std Bold Slanted
- Bell Gothic Std Light
- Bell Gothic Std Bold
- Bell Gothic Std Black
- Myriad® Pro Light
- Myriad Pro Light Italic
- Myriad Pro Regular
- Myriad Pro Italic
- Myriad Pro Bold
- Myriad Pro Semibold
- Myriad Pro Semibold Italic
- Myriad Pro Bold Italic
- Myriad Pro Black
- Myriad Pro Black Italic
- Minion® Pro Regular
- Kozuka Mincho® Std Extra Light
- Kozuka Mincho Std Light
- Kozuka Mincho Std Regular
- Kozuka Mincho Std Medium
- Kozuka Mincho Std Bold
- Kozuka Mincho Std Heavy
- Kozuka Gothic™ Std Extra Light
- Minion Pro Italic
- Minion Pro Semibold
- Minion Pro Semibold Italic
- Minion Pro Bold
- Minion Pro Bold Italic
- EuroSign (TTF)
- Kozuka Gothic Std Light
- Kozuka Gothic Std Regular
- Kozuka Gothic Std Medium
- Kozuka Gothic Std Bold
- Kozuka Gothic Std Heavy
- Kozuka Mincho Pro-VI Regular
- Kozuka Gothic Pro Medium
- Adobe Serif MM
- Adobe Sans MM
- Adobe Ming Std Light
- Adobe Song Std Light
- Adobe Myungjo Std Medium
- Adobe Pi Std
- Courier Std
- Courier Std Bold
- Courier Std Bold Oblique
- Courier Std Oblique
- Symbol

# Index

## A


accessing
- LiveCycle database tables 37
- LiveCycle Policy Server via web browser 26
- product information updates 8
- web application, LiveCycle Reader Extensions 30

Active Directory
- account, creating 28
- application policy node 58

adding
- authentication to LiveCycle Policy Server 28
- certificates to trust certificates 67

Adobe Form Server, upgrading to LiveCycle Forms 75
Adobe LiveCycle Assembler
- upgrading 81

Adobe LiveCycle database
- about initializing 9
- connecting JBoss to 52
- creating 36
- initializing 63
- tables, accessing 37

Adobe LiveCycle Document Security
- integrating with LiveCycle Policy Server 69
- JAR file, configuring 42
- manual installation 33
- post-deployment tasks 31
- trust components 17
- trust.xml file 103
- turnkey installation 20
- upgrading 89

Adobe LiveCycle Form Manager
- upgrading 79

Adobe LiveCycle Forms
- upgrading 75

Adobe LiveCycle PDF Generator
- upgrading 86

Adobe LiveCycle Policy Server
- configuring for EJB access 70
- database drivers, installing 43
- database, connecting JBoss to 52
- database, creating 36
- database, initializing 63
- eDirectory support 30
- endorsed directory, creating 44
- files to run Configuration Manager 42
- integrating LiveCycle products with 69
- JAR files for running 43
- JBoss configuration files, copying 50
- manual installation 33

Adobe LiveCycle Policy Server (Continued)
- run-time properties, configuring 26
- sample application 56
- security configuration 55
- turnkey installation 22
- upgrading 95
- using Kerberos with 28

Adobe LiveCycle Print
- upgrading 75

Adobe LiveCycle products
- configuring multiple for integration 70
- documentation resources 7
- information updates 8
- manual deployment 60
- manual installation 33
- turnkey installation 20
- upgrading to 7.2 or 7.2.1 73

Adobe LiveCycle Reader Extensions
- integrating with LiveCycle Policy Server 69
- JAR file, configuring 42
- manual installation 33
- resetting counter for 62
- Rights credentials 15
- setting up users, roles, and login files 41
- trust components 16
- trust.xml file 103
- turnkey installation 20
- upgrading 92
- web application, accessing 30

Adobe LiveCycle services, setting automatic startup 22, 25
Adobe LiveCycle Workflow
- upgrading 81

Adobe LiveCycle Workflow Designer
- upgrading 84

Adobe LiveCycle Workflow, running LiveCycle Policy Server with 43
application policies, creating 55
application servers
- configuring 49
- installing 36
- log files 62
- supported 11
- updating 74

authentication
- JAAS, configuring on JBoss 55
- Kerberos, adding to LiveCycle Policy Server 28
- provider, configuring eDirectory as 30
- SSL credential 66

automatic installation
- using to upgrade LiveCycle products 74

## B

BAM Server
about upgrade utility 85
upgrading for LiveCycle Workflow 84

## C

certificates
about 18
adding to trust certificates 67
obtaining 17
checklists 13
configuration checklists 13
configuration files
JBoss, copying 50
XML, LiveCycle Policy Server 70
Configuration Manager
files for running 42
using during upgrade process 74
using to configure LiveCycle products 44
configuration properties, retrieving for upgrading 76
configuring
*See also* installing
about 9
application server 49
DB2 data source for JBoss 54
DB2 database for concurrent usage 40
DocumentServicesLibrary file 42
eDirectory as authentication provider 30
JAAS authentication on JBoss 55
JBoss properties 51
LiveCycle Policy Server for EJB access 70
LiveCycle Policy Server run-time properties 26
LiveCycle products for deployment 44
LiveCycle products for integration 69
MySQL data source for JBoss 52
Oracle data source for JBoss 53
SQL Server data source for JBoss 53
SSL on JBoss 65
transaction time-out property 51
trust data 17
users, roles, and user groups 41
connecting JBoss to LiveCycle database 52
conventions, file path 6
copying
deployable files 61
JBoss configuration files 50
library files 42
count.dat file, deleting 62
creating
Active Directory account for Kerberos 28
application policies 55
data source files 52
databases 36
endorsed directory for LiveCycle Policy Server 44
SSL credential 66
users and roles 41
credentials
about 18
LiveCycle Document Security 17
LiveCycle Reader Extensions 15
SSL, creating 66
credentials element, trust.xml file 105
CRL element, trust.xml file 106
CRLs, obtaining 17

## D

data source files, creating on JBoss 52
databases
configuring connection for JBoss 52
creating 36
drivers supported 12
drivers, installing 43
initializing 63
operating system supported 13
supported 12
DB2 database
configuring for concurrent usage 40
creating 38
creating data source for JBoss 54
deploying
about 9, 60
multiple LiveCycle products 10
to JBoss 61
deployment
checklists 13
configuring LiveCycle products for 44
digital certificates, obtaining 17
directories
endorsed, creating 44
Global storage 46, 97
JBoss naming convention 60
document transfer performance, increasing 96
documentation resources 7
DocumentServicesLibrary.jar file, copying 42

## E

EAR files
deployable 61
deploying to JBoss 61
eDirectory directory server, configuring support for 30
EJB access, LiveCycle Policy Server 70
endorsed directory, creating 44
error log, viewing 35

## F

files. *See* EAR files, JAR files, library files, *and* product files
fonts
bundled 110
selecting 46

## G

Global storage directory
- about 46
- temporary files 97

## H

hardware requirements 13
Hardware Security Module 17

## I

initializing database 63
installation
- checklists 13
- methods for 10

installing
- *See also* configuring
- about 9
- applications to JBoss 61
- database drivers 43
- JBoss 41
- multiple LiveCycle products 10
- using manual method 33
- using turnkey method 20

integrating
- LiveCycle products with LiveCycle Policy Server 69
- multiple LiveCycle products 70

## J

JAAS authentication, configuring on JBoss 55
jacorb.properties file, modifying 51
JAR files
- copying 42
- to run LiveCycle Policy Server 43

JBoss
- about configuring manually 49
- configuration files, copying 50
- configuring security on 55
- configuring SSL on 65
- connecting to LiveCycle database 52
- deploying to 61
- enabling SSL on 67
- files, deployable 61
- home directory name 60
- installing 41
- log files 62
- properties, configuring 51
- setting up the environment 50
- starting and stopping 49

JBoss service, uninstalling 71
jboss-service file, modifying 68
JDK support 11
JVM heap size, increasing maximum 96

## K

Kerberos, adding to LiveCycle Policy Server 28
key pair 16
keystore
- about 16
- creating 47
- password 65

## L

LDAP
- security 55
- server support 12

library files
- copying 42
- to run LiveCycle Policy Server 43

LiveCycle. *See* Adobe LiveCycle
log files, viewing 35, 62
login files, setting up for LiveCycle Reader Extensions 41

## M

manual installation
- about 10
- application policy, creating 55
- checklist 14
- creating application policy for 56
- removing 72
- running 33

multiple LiveCycle products, installing and deploying 10
MySQL
- configuring data source for JBoss 52
- database, creating 36
- driver, installing 43
- user account 37

MySQL service, uninstalling 71

## N

network requirements, Kerberos 28

## O

operating systems
- database support 13
- supported 11

Oracle database
- creating 37
- creating data source for JBoss 53

## P

PDF support 12
prefs element, trust.xml file 107
private keys 17
product files, removing 71, 72
public keys 17

## R

Rights credentials, obtaining 15
run-time properties, configuring for LiveCycle Policy Server 26

## S

security
    configuring on JBoss 55, 65
    required to run LiveCycle Document Security 17
    setting for offline documents 27
signing trust.xml file 18
software supported 11
SQL Server
    configuring data source for JBoss 53
    database, creating 40
    drivers, installing 43
SSL
    configuring on JBoss 65
    creating SSL credential on JBoss 66
    credential, creating 66
    enabling on JBoss 67
starting and stopping JBoss 49
Sun ONE application policy node 58
system requirements 11

## T

transaction time-out property, configuring 51
trust certificates, adding certificates to 67
trust components 16
trust data, configuring 17
trust.xml file
    about 16, 17
    format and content 103
    signing and validating 18
trustAnchor element, trust.xml file 104
turnkey installation
    about 10
    application policy, creating 55
    checklist 13
    creating application policy for 56
    running 20
    uninstalling 71
    using to upgrade LiveCycle products 74

## U

uninstalling server components 71
upgrading LiveCycle products 11, 73
user accounts
    MySQL 37
    pulling from directory servers 27
    setting up for LiveCycle Reader Extensions 41

## V

verifying
    LiveCycle Document Security installation 31
    trust.xml file 18
viewing log files 35, 62

## W

Watched Folder
    upgrading 81
web application, LiveCycle Reader Extensions 30
web browser support 11

## X

XML configuration file, LiveCycle Policy Server 70